

LAPORAN PRAKTIK MAGANG

PENGEMBANGAN BACK-END APLIKASI PENGAJUAN PROPOSAL BERBASIS  
WEBSITE MENGGUNAKAN FRAMEWORK LARAVEL

LAZISMU DIY

Setyo Nugroho  
6/1 23  
Herman Y.



Oleh:

Dewa Putra Hernanda

2000018270

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS AHMAD DAHLAN

2024

## HALAMAN PERSETUJUAN INSTANSI

Laporan Praktik Magang ini telah disetujui sebagai proses pengajuan seminar bagi

Nama : Dewa Putra Hernanda  
NIM : 2000018270  
Tempat Praktik Magang : Lazismu DIY  
Realisasi Waktu Pelaksanaan : 3 Bulan

Yogyakarta, 29 Desember 2023

Menyetujui,

The image shows a handwritten signature in black ink over a printed logo. The logo consists of the word "Lazismu" in a stylized font, with a yellow sun-like symbol to the right of the text.

Marzuki, S.E

**HALAMAN PENGESAHAN**

**PRAKTIK MAGANG**

**PENGEMBANGAN BACK-END APLIKASI PENGAJUAN PROPOSAL BERBASIS WEBSITE  
MENGUNAKAN FRAMEWORK LARAVEL**


**LAZISMU DIY**

**DEWA PUTRA HERNANDA**

**2000018270**


**PEMBIMBING**

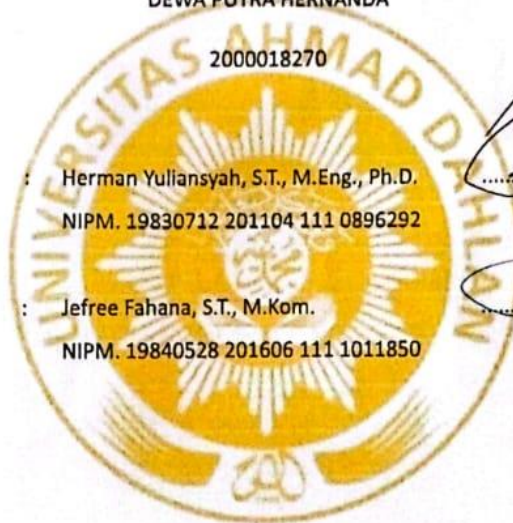
**: Herman Yuliansyah, S.T., M.Eng., Ph.D.  
NIPM. 19830712 201104 111 0896292**

 26/01/24

**PENGUJI**


**: Jefree Fahana, S.T., M.Kom.  
NIPM. 19840528 201606 111 1011850**

 26/01/24



**Yogyakarta, 26 Januari 2024**

**Kaprodi S1 Informatika**

 26/01/24  
**Dr. Murinto, S.Si., M.Kom**

**NIPM. 19730710 200409 111 0951298**

## KATA PENGANTAR

Bersyukur kepada Allah SWT Tuhan Yang Maha Esa atas limpahan rahmat dan petunjuk-Nya, yang telah memberikan banyak kesempatan kepada penulis untuk berhasil menyelesaikan praktik magang yang berjudul "Pengembangan *Back-End* Aplikasi Pengajuan Proposal Berbasis *Website* Menggunakan *Framework Laravel* di Lazismu DIY" Kegiatan magang ini merupakan bagian dari kurikulum pada mata kuliah praktik magang serta merupakan salah satu persyaratan untuk dapat mencapai kelulusan dalam program studi S1 Informatika di Universitas Ahmad Dahlan Yogyakarta.

Selama periode tiga bulan, mulai dari 25 September 2023 hingga 23 Desember 2023, penulis telah menjalankan praktik magang dengan fokus pada pengembangan bagian *back end* dalam pembuatan *website* pengajuan proposal di Lazismu DIY. Laporan praktik magang ini disusun berdasarkan pengalaman dan kontribusi penulis selama masa magang berlangsung. Penyelesaian laporan ini tidak mungkin terwujud tanpa dukungan, semangat, serta bimbingan yang diberikan oleh berbagai pihak, baik dalam bentuk materi maupun non-materi. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT yang telah melimpahkan rahmat, kesehatan, dan mental yang cukup kuat sehingga penulis dapat melaksanakan program praktik magang dan menyelesaikan penyusunan laporan tanpa halangan apapun.
2. Bapak Guntur Maulana Zamroni, B.Sc., M.Kom. selaku Dosen Pengampu kelas praktik magang.
3. Bapak Herman Yuliansyah, S.T., M.Eng., Ph.D. selaku Dosen Pembimbing praktik magang.
4. Bapak Marzuki, S.E. selaku *Manager* Lazismu DIY serta sebagai pembimbing lapangan praktik magang.
5. Riyan Adi Saputro dan Dyah Rossi sebagai rekan tim magang.
6. Sebagai ungkapan terimakasih, laporan praktik magang ini kepada kedua orang tua tercinta, yang selalu menjadi penyemangat dan sandaran terkuat selama ini, yang tiada hentinya selalu memberikan *support*, doa dan motivasi dengan penuh keikhlasan yang tak terhingga.

Perlu disadari bahwa dengan segala keterbatasan, laporan magang ini masih jauh dari sempurna. Sehingga kritikan dan masukan yang membangun sangat penulis harapkan demi kesempurnaan laporan ini. Semoga laporan ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Yogyakarta, 8 Januari 2023



DEWA PUTRA HERNANDA

## DAFTAR ISI

|   |           |
|---|-----------|
| HALAMAN PERSETUJUAN INSTANSI .....  | ii        |
| HALAMAN PENGESAHAN .....  | iii       |
| KATA PENGANTAR.....   | ii        |
| DAFTAR ISI .....  | iv        |
| DAFTAR GAMBAR.....  | vi        |
| DAFTAR TABEL.....   | 1         |
| <b>BAB I .....</b>  | <b>2</b>  |
| A. Latar Belakang.....  | 2         |
| B. Identifikasi Masalah .....   | 3         |
| C. Batasan Masalah .....  | 3         |
| D. Rumusan Masalah .....  | 3         |
| E. Tujuan Praktik Magang.....   | 3         |
| F. Manfaat Praktik Magang.....  | 4         |
| <b>BAB II .....</b>   | <b>5</b>  |
| A. Umum.....  | 5         |
| 1. Latar Belakang Lazizmu .....   | 5         |
| 2. Visi dan Misi .....  | 6         |
| 3. Prinsip.....   | 6         |
| 4. Alamat .....   | 7         |
| 5. Kontak .....   | 7         |
| B. Struktur Organisasi Ditempat Magang.....                                 | 8         |
| C. Sumber Daya Manusia dan Sumber Daya Fisik Di Lokasi Magang.....          | 8         |
| 1. Sumber Daya Manusia .....  | 8         |
| 2. Sumber Daya Fisik .....  | 9         |
| D. Proses Bisnis Saat Ini yang Berhubungan Dengan Tema Praktik Magang ..... | 11        |
| <b>BAB III .....</b>  | <b>12</b> |
| A. Lokasi Praktik Magang, Alamat, Kontak.....                               | 12        |
| B. Rencana Observasi .....  | 12        |
| C. Rancangan Jadwal Kegiatan Magang .....                                   | 13        |
| <b>BAB IV.....</b>  | <b>16</b> |
| A. Hasil Observasi Magang .....   | 16        |
| 1. Proses Bisnis .....  | 16        |
| 2. Rincian Pekerjaan.....   | 17        |

|   |           |
|---|-----------|
| B. Pembahasan Magang .....                      | 18        |
| 1. Problem yang ditemukan di tempat magang..... | 18        |
| 2. Analisis terhadap hasil observasi .....      | 18        |
| 3. Capaian Magang.....                          | 24        |
| 4. Keberlanjutan.....                           | 63        |
| <b>BAB V.....</b>                               | <b>64</b> |
| A. Kesimpulan.....                              | 64        |
| B. Saran.....                                   | 64        |
| <b>LAMPIRAN.....</b>                            | <b>65</b> |

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 Struktur Organisasi Lazismu DIY .....                              | 8  |
| Gambar 2.2 Flowchart penyaluran proposal pengajuan program lazismu .....      | 11 |
| Gambar 3. 1 Lokasi Praktik Magang .....                                       | 12 |
| Gambar 4. 1 Proses Bisnis Pengajuan Proposal.....                             | 16 |
| Gambar 4.2 Gambaran pembagian jobdesk antar developer .....                   | 24 |
| Gambar 4. 3 Tahapan Metode Agile .....  | 26 |
| Gambar 4.4 Entity Relationship Diagram (ERD) .....                            | 27 |
| Gambar 4.5 ERD bagian Pengelola .....   | 27 |
| Gambar 4.6 ERD Bagian Pemohon .....   | 28 |
| Gambar 4.7 ERD Bagian Proposal.....   | 29 |
| Gambar 4.8 ERD Persyaratan Proposal.....                                      | 30 |
| Gambar 4.9 Use Case Diagram .....   | 32 |
| Gambar 4. 10 Activity diagram update profile .....                            | 33 |
| Gambar 4. 11 Activity diagram bagian view history.....                        | 33 |
| Gambar 4. 12 Activity diagram bagian tracking proposal .....                  | 34 |
| Gambar 4. 13 Activity diagram bagian buat akun pemohon.....                   | 34 |
| Gambar 4. 14 Activity diagram bagian upload proposal .....                    | 35 |
| Gambar 4. 15 Activity diagram verifikasi proposal front office.....           | 36 |
| Gambar 4. 16 Activity diagram bagian verifikasi proposal manager.....         | 37 |
| Gambar 4. 17 Activity diagram bagian verifikasi proposal badan pengurus.....  | 37 |
| Gambar 4. 18 Activity diagram bagian program.....                             | 38 |
| Gambar 4. 19 Activity diagram bagian upload proposal .....                    | 39 |
| Gambar 4.20 Activity diagram buat akun pemohon .....                          | 40 |
| Gambar 4.21 Activity diagram buat akun pengelola .....                        | 40 |
| Gambar 4.22 Listing code setting database .....                               | 41 |
| Gambar 4.23 Hasil migrations .....  | 41 |
| Gambar 4.24 Listing code setting seeders.....                                 | 42 |
| Gambar 4. 25 Hasil setelah seeders dijalankan.....                            | 43 |
| Gambar 4. 26 Listing code setting model .....                                 | 43 |
| Gambar 4. 27 Listing code setting mail .....                                  | 44 |
| Gambar 4. 28 Hasil ketika user menerima email dari aplikasi .....             | 45 |
| Gambar 4. 29 Listing code setting route .....                                 | 45 |
| Gambar 4. 30 Listing code fitur tracking proposal .....                       | 46 |
| Gambar 4. 31 Listing code fitur login .....                                   | 46 |
| Gambar 4.32 Listing code fitur logout.....                                    | 47 |
| Gambar 4. 33 Listing code fitur buat akun .....                               | 47 |
| Gambar 4. 34 Listing code verifikasi akun yang dibuatkan oleh pengelola.....  | 48 |
| Gambar 4. 35 Listing code verifikasi akun yang dibuat sendiri oleh user ..... | 48 |



|  |    |
|--|----|
| Gambar 4. 36 Listing code untuk melakukan permintaan lupa password .....       | 49 |
| Gambar 4. 37 Listing code untuk menerapkan fungsi ganti password.....          | 49 |
| Gambar 4. 38 Listing code untuk memberikan notifikasi email dari aplikasi..... | 50 |
| Gambar 4. 39 Listing code untuk upload proposal.....                           | 51 |
| Gambar 4. 40 Listing code untuk fitur verifikasi proposal.....                 | 52 |
| Gambar 4. 41 Listing code fungsi riwayat proposal.....                         | 53 |
| Gambar 4. 42 Listing code read data pengelola .....                            | 53 |
| Gambar 4. 43 Listing code create pengelola .....                               | 54 |
| Gambar 4. 44 Listing code delete pengelola .....                               | 54 |
| Gambar 4. 45 Listing code update data pengelola .....                          | 55 |
| Gambar 4. 46 Listing code read kantor .....                                    | 55 |
| Gambar 4. 47 Listing code delete kantor.....                                   | 55 |
| Gambar 4. 48 Listing code update kantor .....                                  | 56 |
| Gambar 4. 49 Listing code create kantor.....                                   | 56 |
| Gambar 4. 50 Nama responden .....  | 58 |
| Gambar 4. 51 Jabatan / role responden .....                                    | 58 |
| Gambar 4. 52 Nilai kepuasan user terhadap aplikasi .....                       | 58 |



## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2.1 Sumber Daya Manusia Pengelola Lazismu DIY ..... | 8  |
| Tabel 2.2 Sumber Daya Manusia Staff Lazismu DIY .....     | 9  |
| Tabel 3. 1 Rancangan Jadwal Kegiatan Praktik Magang ..... | 14 |
| Tabel 4. 1 Daftar pertanyaan .....                        | 57 |
| Tabel 4. 2 Data hasil pengujian aplikasi .....            | 59 |
| Tabel 4. 3 Data hasil system usability scale .....        | 59 |
| Tabel 4. 4 Jadwal realisasi kegiatan magang.....          | 61 |

## PENDAHULUAN

### A. Latar Belakang

Dalam upaya untuk meningkatkan keterampilan Mahasiswa adalah dengan cara melalui pelaksanaan praktik magang. Di Universitas Ahmad Dahlan, khususnya pada program studi S1 Informatika, Mahasiswa wajib menjalani kegiatan praktik magang pada semester 7 sebagai salah satu persyaratan kelulusan. Pelaksanaan praktik magang berlangsung minimal 3 bulan sesuai dengan ketentuan yang telah ditetapkan. Dipilihnya Lazismu DIY sebagai tempat untuk menjalani praktik magang dikarenakan tempat tersebut sedang membutuhkan sumber daya manusia untuk membantu dalam pemecahan masalah pengajuan proposal.

Tidak hanya itu, alasan dipilihnya Lazismu DIY sebagai tempat praktik magang adalah karena perusahaan ini merupakan salah satu entitas di sektor sosial yang memiliki kantor di beberapa lokasi di Indonesia, termasuk di Kota Yogyakarta. Organisasi ini adalah bagian dari Muhammadiyah dan mengutamakan berbagai kegiatan di bidang keagamaan, sosial, kesehatan, dan pendidikan. Dalam pengelolaan kegiatan Lazismu bergerak sebagai pengumpul dana atau sumber dana yang dimana dari kegiatan itu perlu adanya pengecekan proposal pengajuan dana.

Proses pengajuan proposal dana awalnya dikelola secara manual dengan menggunakan alat-alat di Google seperti spreadsheet untuk mencatat proposal yang diterima, diverifikasi, diajukan, didisposisikan, disetujui, dan ditolak yang dimana itu memiliki rentang waktu yang panjang agar dapat tersimpan data status pengajuan proposal tersebut. Oleh karena itu, Lazismu meminta tim untuk membuat *website* pengajuan proposal dengan tujuan meningkatkan efisiensi dalam proses pengajuan proposal.

Pengembangan aplikasi pengajuan proposal di Lazismu DIY dilakukan secara tim oleh 3 anggota, yaitu system analyst, front-end web developer, dan back-end web developer. Back-end web developer bertanggung jawab untuk mengatur konfigurasi system dan database dari aplikasi yang dikembangkan.

## **B. Identifikasi Masalah**

Berdasarkan latar belakang yang telah disusun maka dapat dibuat identifikasi masalah dari praktik magang ini yaitu sebagai berikut.

1. Sistem manual membuat pemohon sulit untuk melacak status pengajuan proposal secara real-time.
2. Proses manual menyulitkan pelaporan dan analisis data. Kesulitan ini dapat menghambat kemampuan organisasi untuk memahami tren, membuat keputusan strategis, atau melakukan evaluasi kinerja.
3. Perlu adanya sistem untuk memudahkan proses dalam pengelolaan pengajuan proposal yang efisien.

## **C. Batasan Masalah**

Adapun batasan masalah pada kegiatan praktik magang ini dengan membuat aplikasi pengajuan proposal berbasis *website*, yaitu diantaranya.

1. Pengembangan aplikasi proposal LAZISMU DIY dengan menggunakan bahasa pemrograman *PHP* dengan *framework laravel* serta *bootstrap* untuk *styleing website*.
2. Fitur yang dibuat mencakup alur pengajuan proposal dari beberapa pengguna seperti pemohon, *front office*, *manager*, program, badan pengurus, dan admin.

## **D. Rumusan Masalah**

Berdasarkan identifikasi masalah tersebut, maka terdapat beberapa rumusan masalah yaitu diantaranya:

1. Bagaimana cara melakukan pengembangan *back-end* aplikasi pengajuan proposal LAZISMU DIY dengan menggunakan metode pengembangan *agile*.

## **E. Tujuan Praktik Magang**

Tujuan dilaksanakannya kegiatan praktik magang yaitu sebagai berikut.

1. Dapat mengetahui tentang cara kerja dan juga sistem kerja di dunia industri yang di terapkan di Lazismu DIY.
2. Untuk mengetahui cara melakukan pengembangan sistem *website* pengajuan proposal dengan menggunakan *framework laravel*.
3. Menambah pengalaman dan wawasan mengenai dunia kerja.
4. Menambah *skill* dalam aspek pengembangan sistem *website*.

#### **F. Manfaat Praktik Magang**

Adapun manfaat dilaksanakannya kegiatan praktik magang yaitu sebagai berikut.

- a. Mahasiswa dapat terlatih dan menerapkan keterampilan dasar yang telah diperoleh selama studi ke dalam konteks industri.
- b. Mahasiswa dapat meraih pengalaman yang bermanfaat sebagai persiapan untuk menghadapi tantangan dunia kerja.
- c. Mahasiswa dapat menyelami dan memahami peran seseorang yang bekerja dalam pengembangan *website* untuk pengajuan proposal di Lazismu DIY.

# **BAB I**

## **GAMBARAN INSTANSI**

### **A. Umum**

#### **1. Latar Belakang Lazismu**

LAZISMU adalah lembaga yang mengelola zakat, infaq, wakaf, dan dana kedermawanan lainnya dari berbagai sumber. LAZISMU didirikan oleh PP. Muhammadiyah pada tahun 2002 dan dikukuhkan sebagai Lembaga Amil Zakat Nasional oleh Menteri Agama Republik Indonesia pada November 2002. LAZISMU berperan dalam memberdayakan masyarakat melalui pengelolaan dana tersebut.

LAZISMU didirikan karena Indonesia menghadapi masalah kemiskinan, kebodohan, dan indeks pembangunan manusia yang rendah. Hal ini disebabkan oleh kelemahan tatanan keadilan sosial. Zakat dianggap sebagai solusi untuk mengatasi masalah tersebut.

LAZISMU didirikan sebagai lembaga zakat yang modern dan profesional, dengan tujuan untuk menjadikan zakat sebagai solusi bagi masalah sosial. LAZISMU berkomitmen untuk menjalankan amanah, profesionalitas, dan transparansi dalam pengelolaan zakat, sehingga dapat menjadi lembaga zakat yang terpercaya dan dipercaya oleh masyarakat.

LAZISMU berfokus pada kreativitas dan inovasi dalam mengembangkan program-program pendayagunaan zakat yang dapat menjawab tantangan perubahan dan masalah sosial masyarakat. LAZISMU telah tersebar di seluruh Indonesia, sehingga program-program pendayagunaannya dapat menjangkau seluruh wilayah dengan cepat, fokus, dan tepat sasaran. LAZISMU percaya bahwa zakat dapat berperan penting dalam menyelesaikan masalah sosial dan mendorong keadilan sosial.

## **2. Visi dan Misi**

### **a. Visi**

Menjadi Lembaga Amil Zakat Terpercaya

### **b. Misi**

1. Optimalisasi pengelolaan ZIS yang amanah, profesional, dan
2. transparan.
3. Optimasi pendayagunaan ZIS yang kreatif, inovatif dan produktif
4. Optimalisasi pelayanan donatur.

## **3. Prinsip**

Pengelolaan ZISKA berprinsip:

1. Syariat Islam, artinya dalam menjalankan tugas dan fungsinya, harus berpedoman sesuai syariat Islam, mulai dari tata cara perekrutan pegawai hingga tata cara pendistribbusian ZISKA.
2. Amanah dan integritas, artinya harus menjadi lembaga yang dapat dipercaya, dengan memegang teguh kode etik dan prinsip-prinsip moral.
3. Kemanfaatan, artinya memberikan manfaat yang besar bagi mustahik.
4. Keadilan, artinya mampu bertindak adil, yakni sikap memperlakukan secara setara di dalam memenuhi hak-hak yang timbul berdasarkan perjanjian serta peraturan perundangan yang berlaku.
5. Kepastian hukum, artinya muzaki dan mustahik harus memiliki jaminan dan kepastian hukum dalam proses pengelolaan ZISKA.
6. Terintegrasi, artinya harus dilakukan secara heirarkis sehingga mampu meningkatkan kinerja pengumpulan, pendistribusian dan pendayagunaan dana ZISKA.
7. Akuntabilitas, artinya pengelolaan dana ZISKA harus bisa dipertanggungjawabkan kepada masyarakat dan mudah diakses oleh masyarakat dan pihak lain yang berkepentingan.
8. Profesional, artinya perilaku yang selalu mengedepankan sikap dan Tindakan yang dilandasi oleh tingkat kompetensi, kredibilitas dan komitmen yang tinggi.



9. Transparansi, artinya tindakan menyampaikan informasi secara transparan, konsisten, dan kredibel untuk memberikan layanan yang lebih baik dan lebih cepat kepada pemangku kepentingan.

10. Sinergi, artinya sikap membangun dan memastikan hubungan kerja sama internal yang produktif serta kemitraan yang harmonis dengan para pemangku kepentingan dana ZISKA untuk menghasilkan karya yang bermanfaat dan berkualitas.

Berkemajuan, artinya melakukan sesuatu secara baik dan benar yang berorientasi ke depan.

#### **4. Alamat**

JL. Gedongkuning 130B Yogyakarta 5571

#### **5. Kontak**

T: (0274)377078

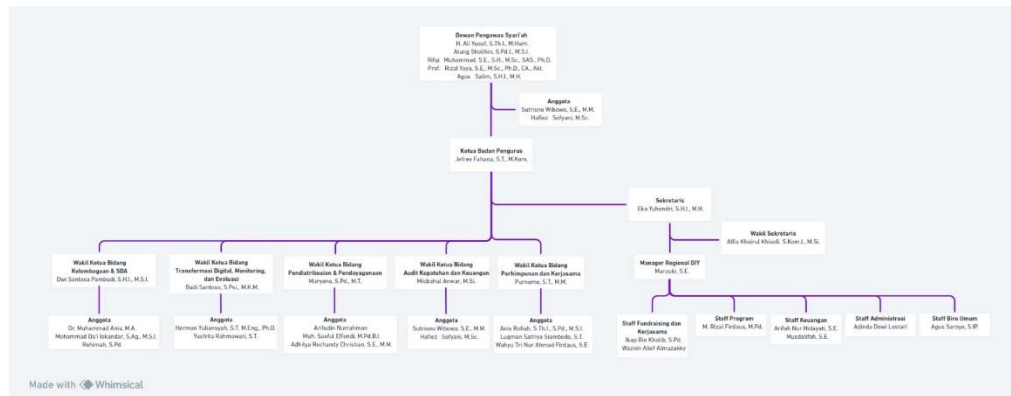
F: (0274)731718

HP: 082138339339

Email : [Lazismudiy@gmail.com](mailto:Lazismudiy@gmail.com)

## B. Struktur Organisasi Ditempat Magang

Struktur organisasi menunjukkan pola hubungan dari suatu organisasi yang memiliki keterkaitan antara satu dengan yang lain. Pada Gambar 2.1 berikut merupakan bagan struktur organisasi pada Lazismu DIY.



Gambar 2.1 Struktur Organisasi Lazismu DIY

## C. Sumber Daya Manusia dan Sumber Daya Fisik Di Lokasi Magang

### 1. Sumber Daya Manusia

Sumber daya manusia (SDM) adalah salah satu faktor yang paling penting bagi keberhasilan suatu organisasi. SDM menggunakan sumber daya lain yang dimiliki organisasi untuk mencapai tujuannya. Lazismu DIY memiliki 22 pegawai yang terdiri dari pengelola dan staf. Pengelola bertanggung jawab atas pengelolaan organisasi secara keseluruhan, sedangkan staf bertanggung jawab atas tugas-tugas tertentu dalam pengelolaan organisasi. Dimana seperti tertera pada Tabel 2.1 merupakan data dari pengelola yang ada di lazismu DIY sedangkan pada Tabel 2.2 merupakan data staff yang ada di lazismu DIY.

Tabel 2.1 Sumber Daya Manusia Pengelola Lazismu DIY

| No. | Pengelola                              | Jumlah  |
|-----|--|---------|
| 1   | Dewan Pengawas Syari'ah                | 4 orang |
| 2   | Badan Pengurus                         | 1 orang |
| 3   | Ketua                                  | 1 orang |
| 4   | Sekretaris                             | 1 orang |
| 5   | Wakil Sekretaris                       | 1 orang |
| 6   | Wakil Ketua Bidang Audit Kepatuhan dan | 1 orang |

|    |   |         |
|----|---|---------|
|    | Keuangan  |         |
| 7  | Anggota   | 2 orang |
| 8  | Wakil Ketua Bidang Perhimpunan dan Kerjasama                      | 1 orang |
| 9  | Anggota   | 3 orang |
| 10 | Wakil ketua Bidang Pendistribusian & Pendayagunaan                | 1 orang |
| 11 | Anggota   | 3 orang |
| 12 | Wakil ketua Bidang Transformasi Digital, Monitoring, dan Evaluasi | 1 orang |
| 13 | Anggota   | 2 orang |
| 14 | Wakil Ketua Bidang Kelembagaan & SDA                              | 1 orang |
| 15 | Anggota   | 3 orang |

*Tabel 2.2 Sumber Daya Manusia Staff Lazismu DIY*

| No. | Nama Bagian                     | Jumlah  |
|-----|---------------------------------|---------|
| 1   | Manager Regional                | 1 orang |
| 2   | Staff Keuangan                  | 2 orang |
| 3   | Staff Administrasi              | 1 orang |
| 4   | Staff Program                   | 1 orang |
| 5   | Staff Biru Umum                 | 1 orang |
| 6   | Staff Fundarising dan Kerjasama | 2 orang |

## **2. Sumber Daya Fisik**

Sumber daya fisik adalah sumber daya yang dapat dilihat dan disentuh, serta dapat digunakan untuk mendukung kegiatan. Salah satu sumber daya fisik di Lazismu DIY adalah fasilitas material yang dimiliki Lazismu DIY. Fasilitas material ini terdiri dari :

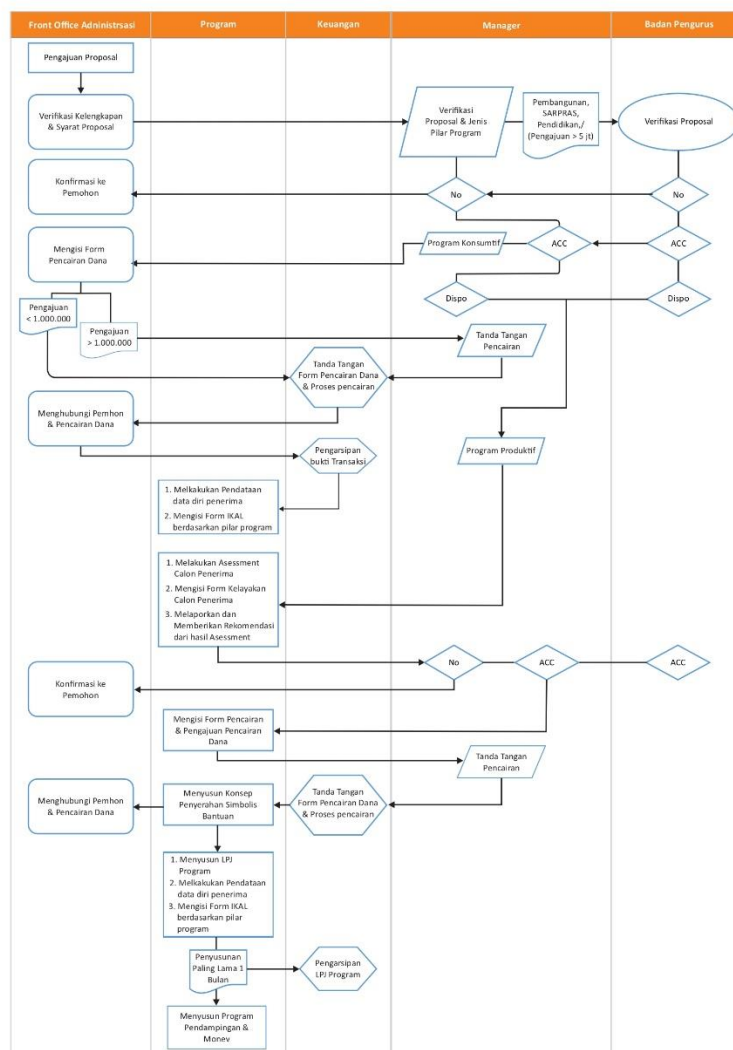
- a) Ruang Lazismu
- b) Komputer
- c) Laptop
- d) Printer

- e) Mobil
- f) Proyektor

#### D. Proses Bisnis Saat Ini yang Berhubungan Dengan Tema Praktik Magang

Proses pengajuan proposal di Lazizmu DIY saat ini masih dilakukan secara manual. Calon penerima bantuan harus datang ke kantor Lazizmu DIY untuk mengajukan proposal. Perubahan status proposal hanya dicatat melalui spreadsheet. Hal ini menyebabkan proses pengajuan proposal menjadi kurang efisien, karena membutuhkan waktu dan tenaga yang lebih banyak, dan dapat dilihat lebih jelas flownya seperti pada gambar 2.2 dibawah.

Flowchart Penyaluran Prgram Lazizmu Berdasarkan Proposal Pengajuan



Gambar 2.2 Flowchart penyaluran proposal pengajuan program lazizmu

## BAB II

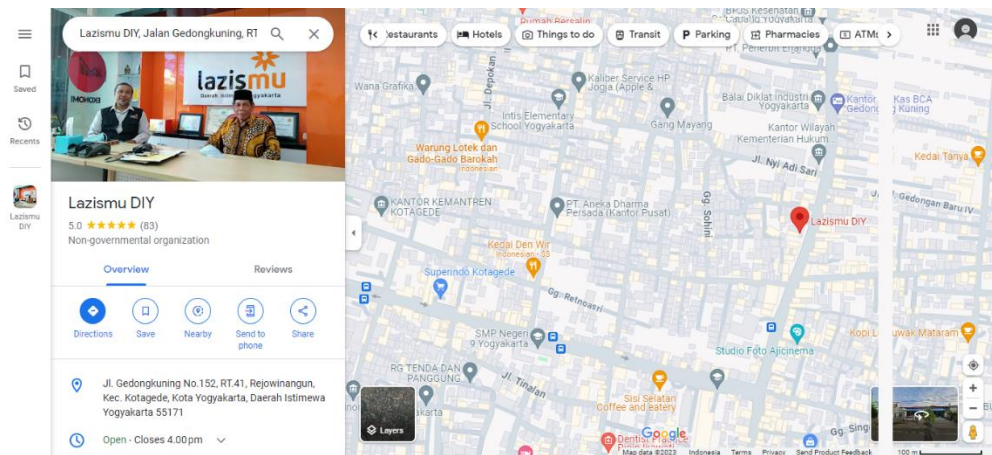
### TAHAPAN KEGIATAN PRAKTIK MAGANG

#### A. Lokasi Praktik Magang, Alamat, Kontak

Praktik magang dilaksanakan di Lazismu DIY, yang beralamat di Jalan Gedongkuning 130B Yogyakarta 5571. Alamat tersebut dapat dilihat pada Gambar 3.1.

Kontak : 082138339339

Email : <http://www.lazismudiy.or.id/>



Gambar 3. 1 Lokasi Praktik Magang

#### B. Rencana Observasi

##### 1. Perencanaan

###### a. Nama Instansi

Nama instansi dari tempat dilaksanakannya praktik magang adalah Lazismu DIY.

###### b. Proses Observasi

Pada hari Sabtu, 30 September 2023, mahasiswa yang melaksanakan praktik magang di Lazismu DIY melakukan kunjungan ke kantor Lazismu DIY. Kunjungan tersebut didampingi oleh dosen pembimbing magang dan dihadiri oleh manager Lazismu DIY. Dalam kunjungan tersebut, mahasiswa dan manager Lazismu DIY berdiskusi mengenai pembuatan sistem pengajuan proposal yang berbasis website.

Hasil dari diskusi tersebut adalah terbentuknya alur sistem secara online, detail pengerjaan website, termasuk pembagian tugas antar anggota magang, dan estimasi waktu pengerjaan.

### **C. Rancangan Jadwal Kegiatan Magang**

Praktik magang dilaksanakan selama 3 bulan, dari tanggal 1 Oktober 2023 hingga 30 Desember 2023. Berikut adalah rincian jadwal praktik magang yang telah dirancang sebelumnya di Lazismu DIY :

Tabel 3. 1 Rancangan Jadwal Kegiatan Praktik Magang

| No | Nama Kegiatan  | Oktober |   |   |   | November |   |   |   | Desember |   |   |   |
|----|--|---------|---|---|---|----------|---|---|---|----------|---|---|---|
|    |  | 1       | 2 | 3 | 4 | 1        | 2 | 3 | 4 | 1        | 2 | 3 | 4 |
| 1  | Pemahaman terhadap kebutuhan system  | ■       |   |   |   |          |   |   |   |          |   |   |   |
| 2  | Pemahaman terhadap <i>framework laravel</i>  | ■       | ■ |   |   |          |   |   |   |          |   |   |   |
| 3  | Pembuatan rancangan <i>relational database management system</i>   | ■       | ■ |   |   |          |   |   |   |          |   |   |   |
| 4  | Implementasi <i>relational database management system</i> , kedalam kode <i>laravel (migrations &amp; seeders)</i> |         |   | ■ | ■ |          |   |   |   |          |   |   |   |
| 5  | Pembuatan fungsionalitas untuk <i>user pemohon</i>   |         |   |   |   | ■        | ■ |   |   |          |   |   |   |
| 6  | Pembuatan fungsionalitas untuk <i>user front office</i> serta perbaikan bug minor                                  |         |   |   |   |          | ■ | ■ |   |          |   |   |   |
| 7  | Pembuatan fungsionalitas untuk <i>user manager</i> serta perbaikan bug   |         |   |   |   |          |   | ■ | ■ |          |   |   |   |



|    |   |  |  |  |  |  |  |  |  |  |  |  |  |
|----|---|--|--|--|--|--|--|--|--|--|--|--|--|
|    | minor   |  |  |  |  |  |  |  |  |  |  |  |  |
| 8  | Pembuatan fungsionalitas untuk <i>user</i> badan pengurus serta perbaikan bug minor |  |  |  |  |  |  |  |  |  |  |  |  |
| 9  | Pembuatan fungsionalitas untuk <i>user</i> program serta perbaikan bug minor        |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Pembuatan fungsionalitas untuk <i>user</i> admin serta perbaikan bug minor          |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Melakukan testing aplikasi ke client  |  |  |  |  |  |  |  |  |  |  |  |  |

# BAB III

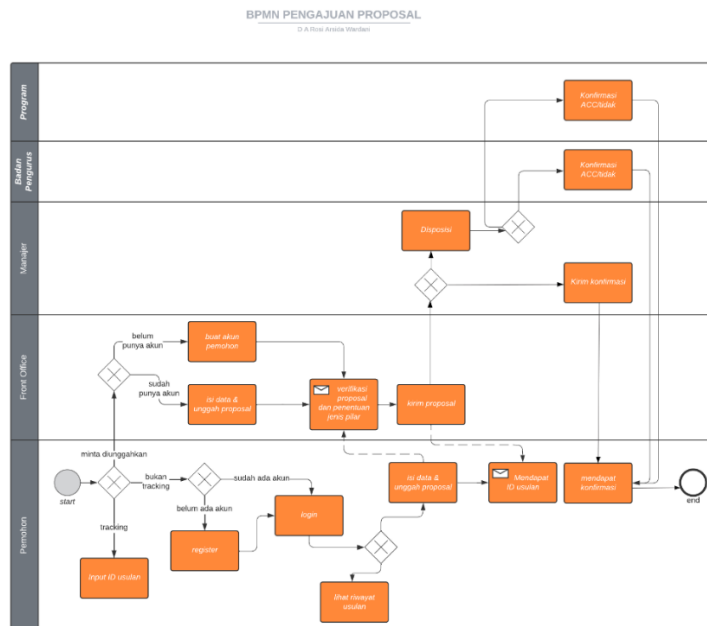
## HASIL PELAKSANAAN PRAKTIK MAGANG

### A. Hasil Observasi Magang

Hasil observasi praktik magang berisikan proses bisnis serta rincian pekerjaan pada instansi pelaksanaan magang.

#### 1. Proses Bisnis

Pada proses bisnis pengajuan proposal yang berlangsung di Lazismu DIY, melakukan pembaharuan pada proses bisnis yang dimana sebelumnya lebih mengarah dengan cara *offline* (harus datang ke kantor), kini dibuat menjadi *online* (tidak harus datang ke kantor). *Website* pengajuan proposal akan menggunakan *framework laravel*. Pada proses bisnis yang telah dirancang, terdapat 5 role/type seperti [**pemohon (pengguna)**], [**front office, manager, badan pengurus, dan program (pengelola kantor)**]. Dengan tugas untuk **pemohon** dapat upload proposal dan melakukan tracking proposal, sedangkan untuk **pengelola kantor** mereka dapat melakukan verifikasi propoasa yang sebelumnya diberikan oleh **pemohon**.



Gambar 4. 1 Proses Bisnis Pengajuan Proposal

## 2. Rincian Pekerjaan

Rincian pekerjaan selama praktik magang di Lazismu DIY yaitu sebagai berikut.

- a. Pengembangan sistem Aplikasi pengajuan proposal berbasis *website*.
  - 1) Mempelajari *framework Laravel*
    - a) Mempelajari arsitektur dalam *framework Laravel*.
    - b) Mempelajari alur program sistem Aplikasi pengajuan proposal.
    - c) Mempelajari fitur-fitur pada *Laravel*, yang dibutuhkan seperti :
      - a. *Database*
      - b. *Eloquent ORM*
      - c. *Security*
      - d. *File Storage*
      - e. *Mail*
      - f. *Routing*
      - g. *Blade Templates*
      - h. *Controllers*
      - i. *Middleware*
      - j. *Validation*
  - 2) Praktik dalam pengembangan sistem Aplikasi pengajuan proposal berbasis *website*
    - a) Melakukan rapat diskusi antara mahasiswa praktik magang dengan direktur utama untuk membahas inti permasalahan yang ingin diselesaikan dalam project aplikasi pengajuan proposal berbasis *website*.
    - b) Melakukan proses pencarian ide pada fitur sistem aplikasi pengajuan proposal berbasis *website*.
    - c) Membuat rancangan *Entity Relationship Diagram (ERD)* dan kebutuhan aplikasi.
    - d) Implementasi *Entity Relationship Diagram (ERD)* kedalam kode *Laravel* (*migrations & seeder*).
    - e) Menambahkan *security* pada pembuatan akun, dimana *user* harus melakukan verifikasi melalui email sebelum dapat login setelah melakukan daftar akun.

- f) Melakukan perancangan *design* UI/UX dari fitur yang akan dibuat.
  - g) Implementasi *project* desain dan fitur pada aplikasi pengajuan proposal berbasis *website* ke dalam kode program *Laravel* (model dan controller).
- 3) Melakukan pengujian aplikasi
- Melakukan proses pengujian untuk dilakukan pemeriksaan program apakah terdapat fitur yang tidak berjalan atau apakah ada *bug* untuk dilakukan perbaikan.
- b. Melakukan analisis sistem pada *project* Aplikasi pengajuan proposal berbasis *website*.
- 1) Menganalisis fitur-fitur yang akan dikembangkan dalam proses pembuatan Aplikasi pengajuan proposal berbasis *website*.
  - 2) Menganalisis kebutuhan sistem Aplikasi pengajuan proposal berbasis *website*.

## **B. Pembahasan Magang**

Adapun penjelasan mengenai praktik magang yang telah dilaksanakan yaitu sebagai berikut.

### **1. Problem yang ditemukan di tempat magang**

Adapun permasalahan yang ditemukan di Lazismu DIY sebagai bahasan topik praktik magang yaitu sebagai berikut.

- a) Proses pengajuan proposal di Lazismu masih secara manual, dan kurang efektif karena prosesnya harus mewajibkan segala karyawan untuk datang langsung ke kantor.
- b) Diperlukan adanya program yang dapat melakukan digitalisasi proses pengajuan proposal, supaya dari pihak pemohon serta karyawan dapat lebih efektif dan lebih mudah untuk saling berbagi informasi.

### **2. Analisis terhadap hasil observasi**

Setelah melakukan observasi terhadap sistem pengajuan proposal, maka terlihat diperlukannya perancangan sebuah aplikasi yang dapat meningkatkan efektivitas dalam pengajuan proposal. Karena pada saat ini

proses dalam pengajuan proposal masih dilakukan secara manual yang membuat kesusahan bagi para pemohon seperti untuk melakukan tracking proposal mereka harus menunggu pesan dari kantor tempat pengajuan proposal mereka, dan bagi karyawan mereka harus menunggu dari *step* sebelumnya memberikan berkas yang dimana terkadang memerlukan proses tidak sedikit, terlebih lagi jika terjadi revisi didalam proposal mereka harus mengulangi prosesnya lagi yang membuat karyawan *front office* menjadi mendapatkan pekerjaan ekstra. Oleh karena itu, diperlukan solusi inovatif berupa aplikasi pengajuan proposal berbasis *website* yang nantinya dapat memberikan kemudahan. Sistem ini tidak hanya akan mempermudah dalam proses pengecekan proposal namun juga akan mempermudah dalam melakukan verifikasi proposal bagi seluruh karyawan dari *role front-office, manager, program*, dan badan pengurus, serta dapat memudahkan bagi para staf untuk melakukan *tracking* berapa proposal yang telah diajukan di kantor tersebut, dan berapa realisasi dana yang telah di-setujui.

Dalam tahap implementasi pembuatan sistem aplikasi pengajuan proposal berbasis *website*, pemahaman terhadap dasar-dasar pengembangan sistem tersebut menjadi krusial. Salah satu hal yang perlu dipahami adalah penggunaan *framework Laravel*, di mana pengetahuan mengenai arsitektur *Laravel*, seperti MVC, menjadi sangat penting. MVC atau *model view control* merupakan sebuah pola arsitektur yang dirancang untuk memudahkan para *developer* dalam pengembangan aplikasi dimana nantinya antara kode untuk mengatur *interface (HTML, CSS, JavaScript)*, dan logika pemrograman (*PHP, MySql*) akan dibuat berbeda. Dengan memahami konsep dari arsitektur ini akan memberikan landasan yang kokoh dalam proses pengembangan sistem Aplikasi pengajuan proposal berbasis *website* dengan menggunakan *framework laravel*.

a) Deskripsi sistem yang dibangun

Sistem aplikasi pengajuan proposal berbasis *website* memiliki fungsi untuk upload proposal bagi *user* dengan *type/role* sebagai pemohon dan *front office* (mewakili pemohon untuk upload proposal), verifikasi proposal bagi *user* dengan *type/role* sebagai *front office, manager, program*, dan badan pengurus. Selain itu para pengelola lazismu dapat melakukan tracking pengeluaran dana bantuan yang terealisasikan dari proposal yang telah disetujui, serta dapat melihat jumlah proposal yang telah dikirimkan di kantor mereka. Tujuan pengembangan sistem ini adalah untuk melakukan digitalisasi dari proses pengajuan proposal supaya dapat dilakukan secara online dan mempermudah para *user* dalam melakukan pengajuan dan verifikasi proposal. Sistem ini dikembangkan menggunakan bahasa pemrograman *PHP* dengan *framework laravel*, serta menggunakan beberapa *JavaScript*. Dengan pendekatan ini, diharapkan aplikasi pengajuan proposal berbasis *website* dapat memberikan solusi yang lebih efektif serta modern dalam melakukan pengajuan proposal di Lazismu.

b) Analisis kebutuhan perangkat lunak (*software*)

Agar sistem tersebut dapat dikembangkan dan dapat digunakan sebagaimana mestinya, maka diperlukan beberapa *tools* sebagai penunjang dalam pengembangan sistem aplikasi. Adapun kebutuhan perangkat lunak (*software*) yaitu sebagai berikut.

- 1) Sistem Operasi Windows.
- 2) Visual Studio Code untuk olah code.
- 3) Git dan Github untuk melakukan *version control*.
- 4) Brave untuk *debugging*.
- 5) Figma untuk membuat rancangan aplikasi.

c) Analisis kebutuhan perangkat keras (*hardware*)

Selain *tools* yang dibutuhkan untuk pengembangan sistem, perangkat lunak (*hardware*) juga menjadi salah satu kebutuhan dalam keberhasilan pengembangan sebuah sistem. Adapun kebutuhan perangkat lunak (*hardware*) yang digunakan dalam pengembangan sistem Aplikasi pengajuan proposal berbasis *website* yaitu sebagai berikut.

- 1) Seperangkat komputer

- 2) Laptop MSI GF63 10SC
  - 3) Ram 10 GB
- d) Analisis kebutuhan sistem aplikasi
- 1) Fitur login dan logout aplikasi
  - 2) Fitur registrasi akun
  - 3) Fitur upload proposal
  - 4) Fitur verifikasi proposal
  - 5) Fitur laporan proposal (dashboard)
  - 6) Fitur generate kode tracking
  - 7) Fitur riwayat proposal
  - 8) Fitur tracking proposal
  - 9) Fitur kelola akun
  - 10) Fitur kelola kantor
- e) Analisis kebutuhan fungsional
- 1) pemohon harus dapat melakukan tracking proposal
  - 2) pemohon harus dapat melakukan login
  - 3) pemohon harus dapat melakukan register
  - 4) pemohon harus dapat melihat riwayat usulan
  - 5) pemohon harus dapat melakukan CRUD proposal selama status proposal masih draft
  - 6) pemohon harus dapat melakukan pengunggahan proposal
  - 7) pemohon harus dapat melihat id usulan
  - 8) pemohon harus dapat melihat profil pribadi
  - 9) pemohon harus dapat mengedit profil pribadi
  - 10) pemohon harus dapat melakukan logout
  - 11) front office harus dapat melakukan tracking proposal
  - 12) front office harus dapat melakukan login
  - 13) front office harus dapat melakukan CRUD akun pemohon
  - 14) front office harus dapat melihat daftar pemohon
  - 15) front office harus dapat melakukan pencarian
  - 16) front office harus dapat mengisi data unggah proposal
  - 17) front office harus dapat melihat daftar proposal
  - 18) front office harus dapat melakukan unggah proposal

- 19) front office harus dapat meng-update status sesuai tidaknya proposal
- 20) front office harus dapat meng-update jenis pilar proposal
- 21) front office harus dapat mengirim proposal ke manager
- 22) front office harus dapat melihat status usulan secara real-time
- 23) front office harus dapat melakukan logout
- 24) manajer harus dapat melakukan login
- 25) manajer harus dapat melihat proposal yang perlu di ACC
- 26) manajer harus dapat meng-update status ACC tidaknya proposal apabila proposal tidak dikirim ke badan pengurus
- 27) manajer harus dapat mengirim proposal ke badan pengurus
- 28) manajer harus dapat melakukan pencarian
- 29) manajer harus dapat melakukan logout
- 30) badan pengurus harus dapat melakukan login
- 31) badan pengurus harus dapat melihat proposal yang perlu di ACC
- 32) badan pengurus harus dapat meng-update status ACC tidaknya proposal
- 33) badan pengurus harus dapat melakukan pencarian
- 34) badan pengurus harus dapat melakukan logout
- 35) sistem harus dapat melakukan tracking proposal
- 36) sistem harus dapat menampilkan form login
- 37) sistem harus dapat melakukan submit data login
- 38) sistem harus dapat menampilkan form register
- 39) sistem harus dapat melakukan submit data register
- 40) sistem harus dapat menyimpan riwayat login (remember me)
- 41) sistem harus dapat menampilkan riwayat usulan proposal pemohon
- 42) sistem harus dapat melakukan CRUD proposal selama status proposal masih draft
- 43) sistem harus dapat melakukan submit proposal
- 44) sistem harus dapat men-generate id usulan
- 45) sistem harus dapat menampilkan id usulan
- 46) sistem harus dapat menampilkan profil pemohon



- 47) sistem harus dapat melakukan update profil pemohon
- 48) sistem harus dapat menampilkan panduan penggunaan website
- 49) sistem harus dapat melakukan logout
- 50) sistem harus dapat melakukan CRUD akun pemohon
- 51) sistem harus dapat menampilkan daftar pemohon
- 52) sistem harus dapat melakukan pencarian
- 53) sistem harus dapat menampilkan form unggah proposal
- 54) sistem harus dapat menampilkan daftar proposal
- 55) sistem harus dapat melakukan unggah proposal
- 56) sistem harus dapat melakukan update status sesuai tidaknya proposal
- 57) sistem harus dapat melakukan update jenis pilar proposal
- 58) sistem harus dapat melakukan pengiriman proposal ke manager
- 59) sistem harus dapat menampilkan status usulan secara real-time
- 60) sistem harus dapat mengirim konfirmasi usulan ke pemohon
- 61) sistem harus dapat menampilkan proposal yang perlu di ACC
- 62) sistem harus dapat meng-update status ACC tidaknya proposal apabila proposal tidak dikirim ke badan pengurus
- 63) sistem harus dapat mengirim proposal ke badan pengurus

f) Analisis kebutuhan non-fungsional

- 1) Tingkat kesalahan pengguna yang mengirimkan detail proposal di halaman proposal tidak boleh melebihi 10%;
- 2) Tulisan yang ditampilkan aplikasi harus mudah dibaca dan ringkas dan on-point;
- 3) Desain interface haruslah tidak membingungkan pengguna (simplen namun tepat sasaran).
- 4) Sistem harus tersedia untuk pengguna, 99,98%
- 5) setiap bulan selama jam kerja.
- 6) Pengguna harus login dengan benar
- 7) terlebih dahulu untuk dapat mengakses fitur
- 8) lengkap
- 9) Sistem harus beroperasi tanpa kegagalan
- 10) (sesuai skenario) dalam 95% use case dalam

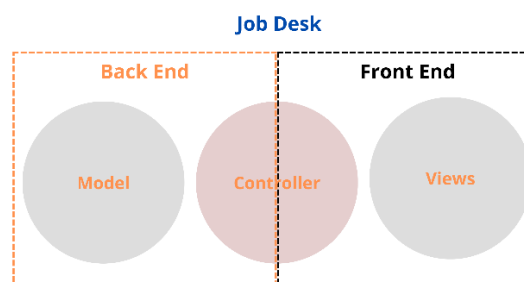
- 11) sebulan
- 12) Sistem melakukan verifikasi ulang apabila ingin mengupdate dan menghapus data
- 13) Sistem dapat diakses dengan mudah, baik di
- 14) mobile maupun PC.
- 15) Kebutuhan akan respon, keluaran, timing dari
- 16) sistem memadai
- 17) Sistem harus cukup terukur untuk mendukung
- 18) 50 kunjungan pada saat yang sama dengan
- 19) tetap menjaga kinerja optimal

### 3. Capaian Magang

Setelah melaksanakan praktik magang selama 3 bulan pada 1 Oktober 2023 hingga 30 Desember 2023 sebagai *Back End Web Developer* di Lazismu DIY. Adapun capaian magang yang diperoleh sebagai berikut.

#### a) Pembagian *Jobdesk* Antara Anggota Team

Dalam pengembangan project aplikasi pengajuan proposal berbasis *website* dikerjakan oleh 3 orang didalam satu team. Dimana dalam pembagian *jobdesk* ada yang berperan sebagai *system analyze*, *front-end web developer*, dan juga *back-end web developer*.



Gambar 4.2 Gambaran pembagian jobdesk antar developer

Gambar 4.2 merupakan gambaran dalam pembagian *jobdesk* untuk masing masing developer. Dimana untuk *Back-end* nantinya akan berperan di balik layar untuk melakukan pengembangan dibagian model, *database*, migrations, dan juga controller. Yang dimana

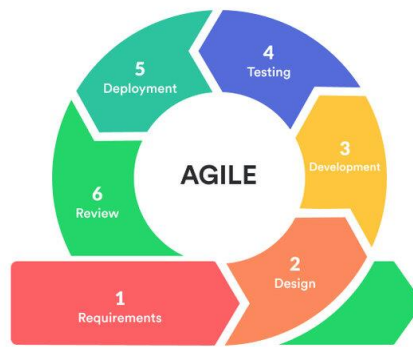
nantinya data yang dibuat oleh *back-end* akan di *consume* oleh *front-end*.

**b) Pengembangan Sistem Aplikasi pengajuan proposal berbasis *website* Menggunakan *Laravel*.**

Adapun beberapa hal yang dibutuhkan dalam pengembangan Aplikasi pengajuan proposal berbasis *website* Menggunakan *Laravel* yaitu sebagai berikut.

1) Metode Perancangan Sistem

Pengembangan pada sistem Aplikasi pengajuan proposal berbasis *website* di Lazismu DIY menggunakan metode agile, dimana Metode ini dipilih karena sesuai dengan karakteristik proyek yang bersifat fleksibel dan dapat disesuaikan dengan perubahan kebutuhan. Proses perencanaan sistem menggunakan metode Agile dimulai dengan perumusan visi dan misi aplikasi. Visi dan misi ini kemudian dijabarkan menjadi kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional adalah fitur-fitur yang harus ada pada aplikasi, sedangkan kebutuhan non-fungsional adalah persyaratan non-teknis yang harus dipenuhi oleh aplikasi, seperti keamanan, kinerja, dan kemudahan penggunaan. Setelah kebutuhan sistem didefinisikan, tim pengembang kemudian melakukan pengembangan prototipe. Prototipe ini digunakan untuk menguji kebutuhan sistem dan mendapatkan umpan balik dari pengguna. Umpan balik tersebut kemudian digunakan untuk menyempurnakan kebutuhan sistem. Proses pengembangan aplikasi kemudian dilakukan secara iteratif, yaitu dengan membagi proyek menjadi beberapa sprint. Setiap sprint berlangsung selama beberapa minggu dan menghasilkan produk yang siap digunakan oleh pengguna. Pada akhir setiap sprint, tim pengembang melakukan pengujian dan validasi untuk memastikan bahwa produk yang dihasilkan sesuai dengan kebutuhan pengguna. Alur kerja dari metode agile dapat dilihat pada Gambar 4.3 berikut.



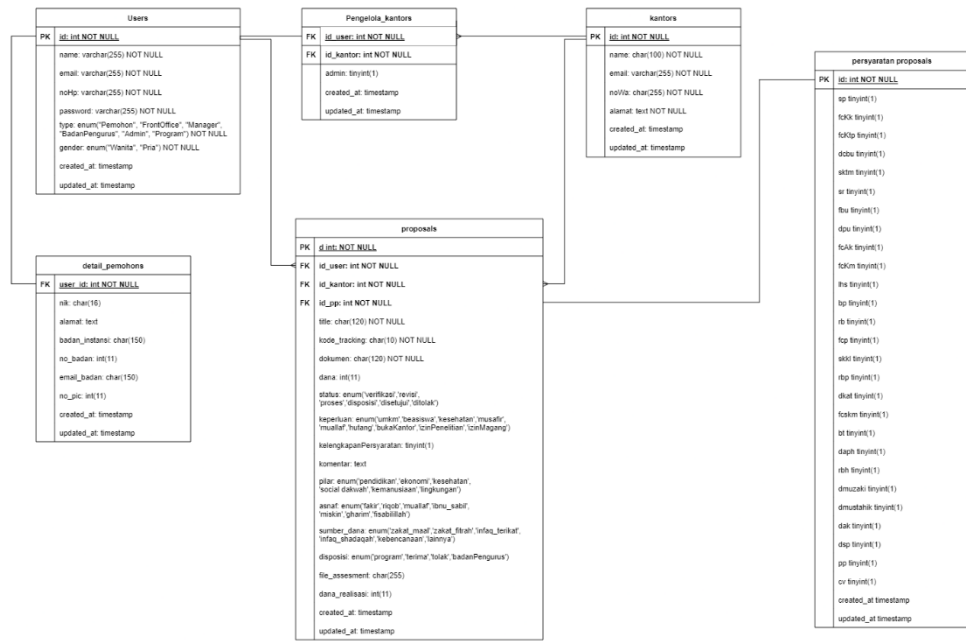
Gambar 4. 3 Tahapan Metode Agile

## 2) Analisis Kebutuhan Sistem

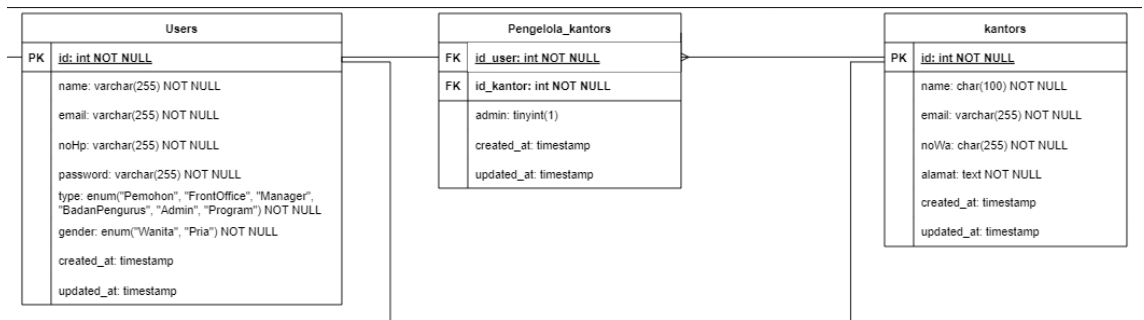
Sistem Aplikasi Pengajuan Proposal ini dirancang sebagai *website* yang akan digunakan oleh “**pemohon**” yang ingin mengajukan proposal, serta “**pengelola**” dimana terdiri dari “**front office, manager, program, badan pengurus**” yang akan melakukan verifikasi berkas pengajuan proposal yang diberikan oleh “**pemohon**”.

### a) Entity Relationship Diagram

*Entity Relationship Diagram (ERD)* adalah representasi grafis dari struktur data dan hubungan antar entitas dalam suatu sistem atau aplikasi. Berikut pada Gambar 4.4 merupakan ERD secara lengkapnya dan akan diberikan penjelasan secara detail dari masing-masing *entity* yang ada.

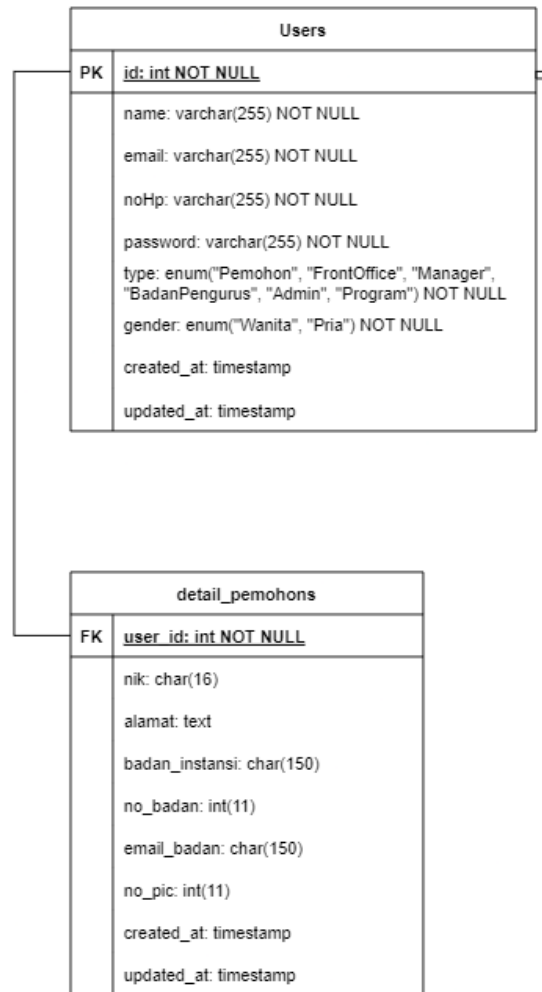


Gambar 4.4 Entity Relationship Diagram (ERD)



Gambar 4.5 ERD bagian Pengelola

Pada gambar 4.5 merupakan erd bagian pengelola, yang akan berguna untuk menyimpan data bagi user pengelola kantor, dimana user yang didaftarkan sebagai pengelola akan memiliki type diantara {front office, manager, badan pengurus, ataupun program}, serta nantinya akan terbuat foreign key (FK) ke table pengelola kantor, yang dimana table pengelola kantor juga akan terhubung dengan primary key dari table kantor, sesuai dengan user pengelola dibuat untuk dimasukkan ke kantor bagian atau cabang yang mana.



Gambar 4.6 ERD Bagian Pemohon

Pada Gambar 4.6 diatas ditunjukkan ERD bagian pemohon yang berguna untuk menyimpan data dari detail pemohon, dimana jika pemohon melakukan regiseter pada halaman *website* pengajuan proposal maka akan otomatis terbuat *user* dengan disertai *forigen key* ke detail\_pemohons, yang dimana isi *field* dari detail\_pemohons merupakan data diri dari pemohon, dan beberapa wajib untuk di-isi sebelum dapat melakukan upload proposal. Data yang wajib untuk di-isi antaranya adalah nik, dan alamat.



Gambar 4.7 ERD Bagian Proposal

Pada Gambar 4.7 diatas merupakan bagian dari Proposal yang berisi data untuk menyimpan detail dari proposal yang diajukan oleh pemohon, yang dimana termuat data-data penting untuk pengajuan proposal, seperti field “status” berguna untuk memberikan informasi progress dari proposal yang diajukan, “kode\_tracking” berguna unuk pemohon supaya dapat melakukan tracking proposal yang diajukan.

| persyaratan proposals |                      |
|-----------------------|----------------------|
| PK                    | id: int NOT NULL     |
|                       | sp tinyint(1)        |
|                       | fcKk tinyint(1)      |
|                       | fcKtp tinyint(1)     |
|                       | dcbu tinyint(1)      |
|                       | sktm tinyint(1)      |
|                       | sr tinyint(1)        |
|                       | fbu tinyint(1)       |
|                       | dpu tinyint(1)       |
|                       | fcAk tinyint(1)      |
|                       | fcKm tinyint(1)      |
|                       | lhs tinyint(1)       |
|                       | bp tinyint(1)        |
|                       | rb tinyint(1)        |
|                       | fcp tinyint(1)       |
|                       | skki tinyint(1)      |
|                       | rbp tinyint(1)       |
|                       | dkat tinyint(1)      |
|                       | fckm tinyint(1)      |
|                       | bt tinyint(1)        |
|                       | daph tinyint(1)      |
|                       | rbh tinyint(1)       |
|                       | dmuzaki tinyint(1)   |
|                       | dmustahik tinyint(1) |
|                       | dak tinyint(1)       |
|                       | dsp tinyint(1)       |
|                       | pp tinyint(1)        |
|                       | cv tinyint(1)        |
|                       | created_at timestamp |
|                       | updated_at timestamp |

Gambar 4.8 ERD Persyaratan Proposal

Pada gambar 4.8 diatas merupakan bagian lanjutan dari bagian proposal, yang dimana *table* ini berguna untuk menyimpan detail dari persyaratan proposal dimana tiap type proposal akan memiliki persyaratannya sendiri sendiri, dengan ditandai dengan input nilai 0 atau 1, sedangkan null digunakan jika ternyata persyaratan tersebut tidak diperlukan di proposal tersebut. Karena tiap tipe proposal memiliki persyaratan yang berbeda-beda.



## b) Use Case Diagram

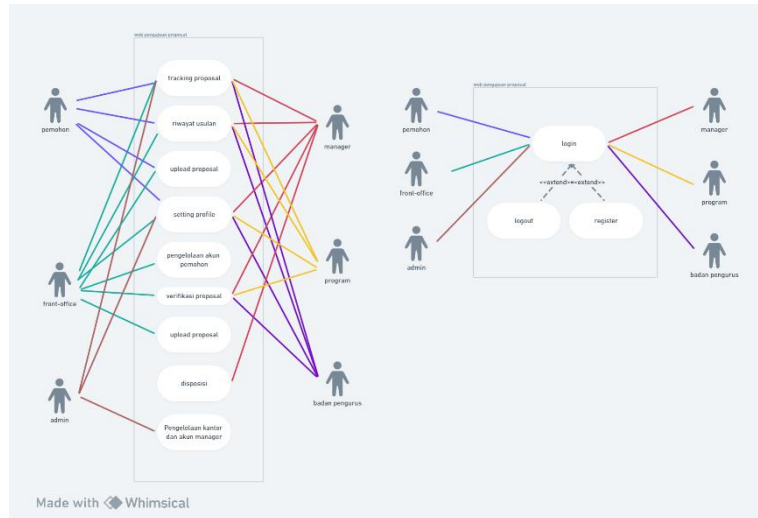
Dalam *use case* diagram yang telah dibuat, terdapat enam aktor utama, yaitu Pemohon, *Front office*, *Manager*, Program, Badan Pengurus, Admin.

Pemohon berperan sebagai *user* yang ingin melakukan pengajuan proposal ke-salah satu kantor lazismu, yang nantinya juga dapat melakukan tracking untuk melihat status dari proposal yang diajukan apakah hasilnya akan disetujui atau masih memerlukan revisi.

*Front office*, *manager*, program, dan badan pengurus berperan sebagai *user* yang akan mengelola di kantor lazismu, yang nantinya akan melakukan proses verifikasi dari berkas yang diajukan ke kantor mereka. Dimana tiap role diatas memiliki cara verifikasinya masing masing, seperti:

- Pemohon, akan melakukan verifikasi dengan mengecek apakah seluruh persyaratan dari proposal yang dikirimkan sudah lengkap atau belum.
- *Manager*, akan melakukan verifikasi apakah proposal tersebut layak untuk disetujui atau tidak, dan jika *manager* merasa tidak mampu untuk menangani proposal tersebut, maka dapat melakukan disposisi ke program ataupun badan pengurus.
- Program, akan melakukan verifikasi dengan memberikan file assesment serta menentukan apakah proposal tersebut layak atau tidak, sebelum nantinya dikembalikan lagi ke *manager*.
- Badan pengurus, akan melakukan verifikasi dengan menentukan apakah proposal tersebut disetujui atau tidak.

Dengan *use case* diagram pada gambar 4.9, dapat dipahami secara komprehensif bagaimana interaksi antara aktor dan sistem terjadi, memungkinkan pemahaman yang lebih baik tentang alur kerja aplikasi presensi tersebut.

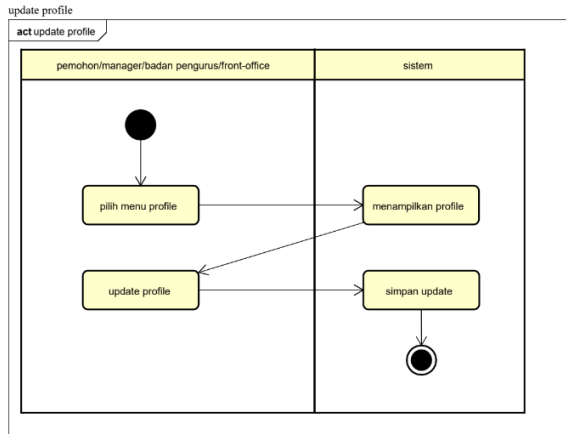


Gambar 4.9 Use Case Diagram

### c) Activity Diagram

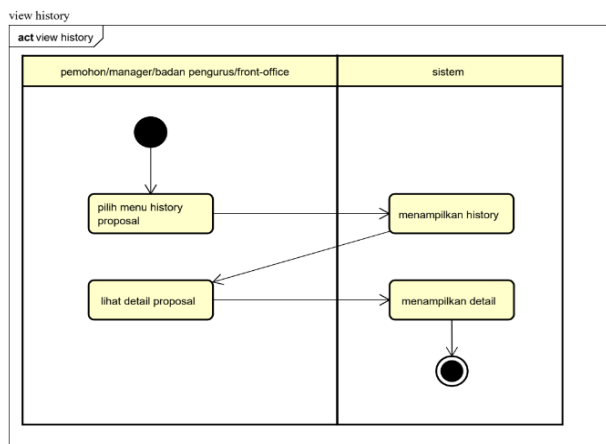
Pada pengembangan sistem aplikasi pengajuan proposal dilakukan pemmembuat beberapa diagram aktivitas (*Activity Diagram*) yaitu profile update, view *history* proposal, tracking proposal, buat akun pemohon, upload proposal, verifikasi proposal front-office, verifikasi proposal *manager*, verifikasi proposal badan pengurus, verifikasi proposal program, upload proposal front-office, register pemohon, dan buat akun pengelola.

Activity diagram bagian profile *update* dapat dilakukan oleh semua role (pemohon, *manager*, badan pengurus, *front office*, program, dan admin), yang dimana *user* hanya perlu masuk ke menu profile lalu setelah *website* mengarahkan ke menu profile maka *user* dapat langsung untuk melakukan *update* data diri seperti nama, email, no Hp, dsb. Berikut activity diagram dapat dilihat pada gambar 4.10.



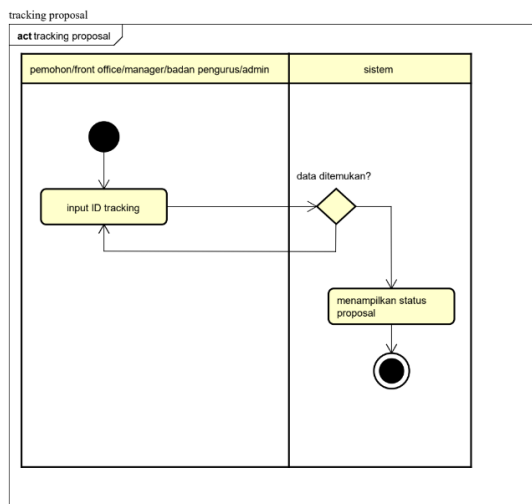
Gambar 4. 10 Activity diagram *update profile*

Activity diagram bagian view *history* proposal dapat dilakukan oleh semua role, namun terdapat sedikit batasan untuk proposal yang ditampilkan seperti, untuk *user* dengan type pemohon maka dia hanya dapat melihat proposal yang pemohon itu upload sendiri, dan untuk *user* dengan type *front office, manager, badan pengurus*, program mereka dapat melihat *history* proposal berdasarkan data proposal yang telah ter-upload di kantor tersebut. Berikut activity diagram dapat dilihat pada gambar 4.11.



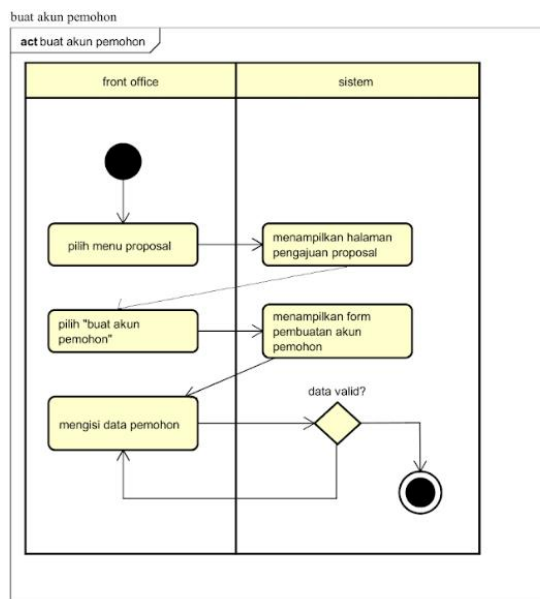
Gambar 4. 11 Activity diagram bagian view *history*

Activity diagram bagian tracking proposal dapat dilakukan oleh semua *user* yang memiliki id tracking, dimana id tracking didapatkan setelah mereka melakukan upload proposal. Berikut activity diagram dapat dilihat pada gambar 4.12.



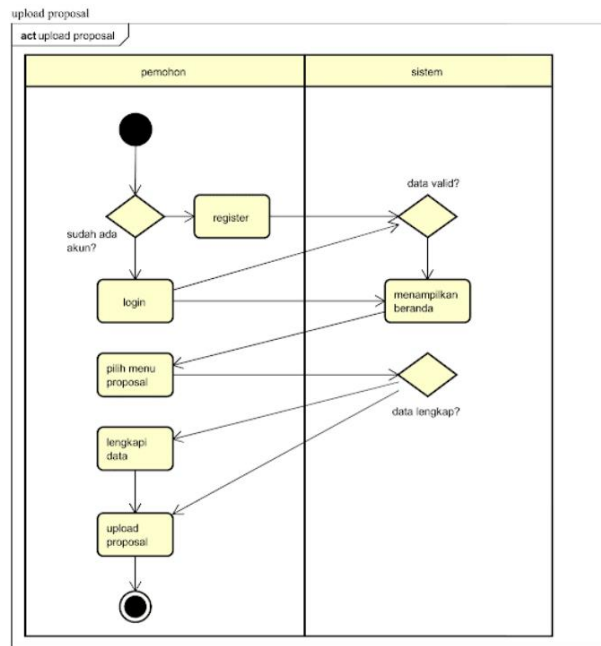
Gambar 4. 12 Activity diagram bagian tracking proposal

Activity diagram bagian buat akun pemohon dapat dilakukan oleh *front office*, yang dimana *front office* dapat membuat akun untuk pemohon jika mereka ingin melakukan upload proposal namun belum memiliki akun. Berikut activity diagram dapat dilihat pada gambar 4.13.



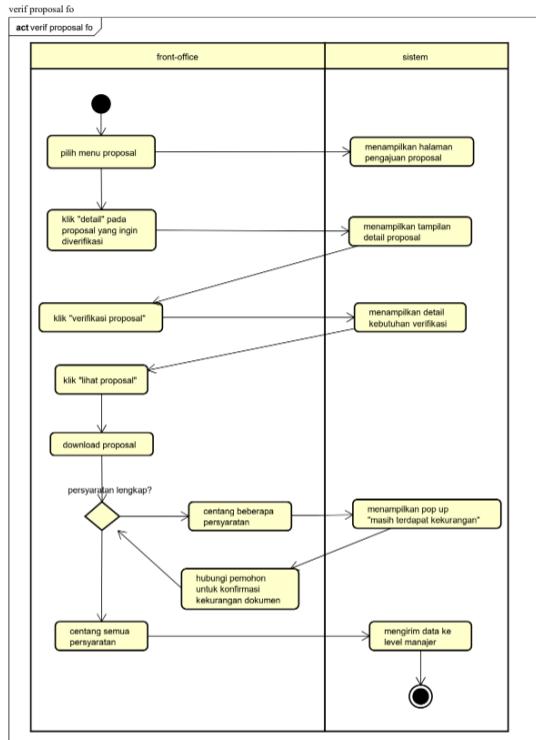
Gambar 4. 13 Activity diagram bagian buat akun pemohon

Activity diagram bagian upload proposal dapat dilakukan oleh pemohon, dimana pemohon yang sudah memiliki akun harus melengkapi terlebih dahulu data dirinya, setelah seluruh data diri lengkap maka *user* tersebut dapat untuk melakukan upload proposal. Berikut activity diagram dapat dilihat pada gambar 4.14.



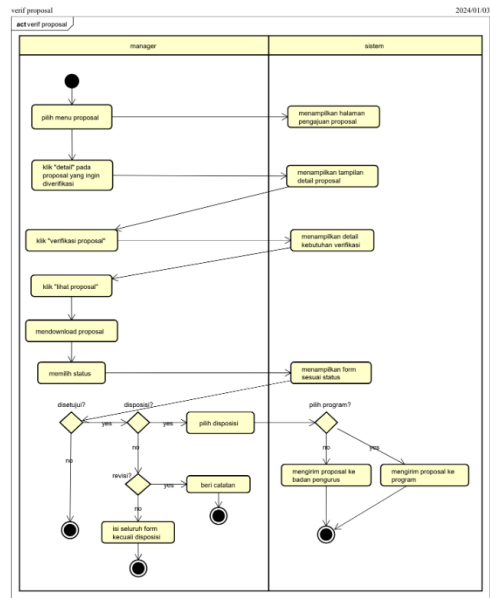
Gambar 4. 14 Activity diagram bagian upload proposal

Activity diagram bagian varifikasi proposal front-office dilakukan oleh *front office*, dimana *front office* akan membuka menu proposal, lalu klik verifikasi pada proposal yang ingin diverifikasi, lalu *front office* akan mengecek persyaratan apakah sudah lengkap atau belum. Berikut activity diagram dapat dilihat pada gambar 4.15.



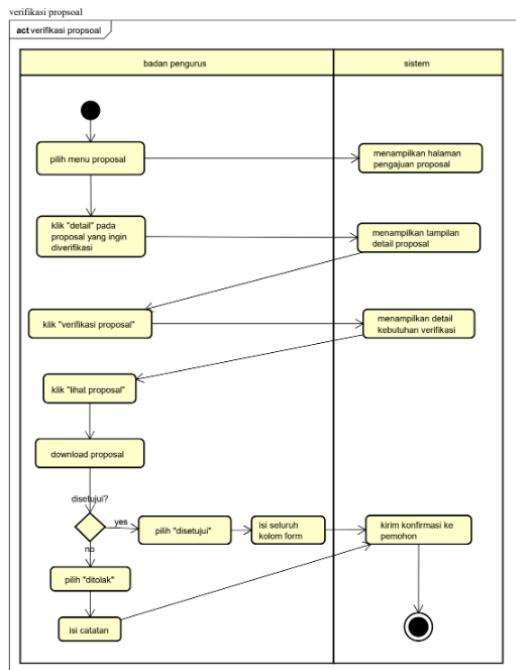
Gambar 4. 15 Activity diagram verifikasi proposal front office

Activity diagram bagian verifikasi proposal manager dilakukan oleh manager, dimana manager akan melakukan verifikasi proposal dengan membuka menu proposal, dan memilih apakah proposal tersebut diterima, ditolak, disposisi, ataupun revisi. Berikut activity diagram dapat dilihat pada gambar 4.16.



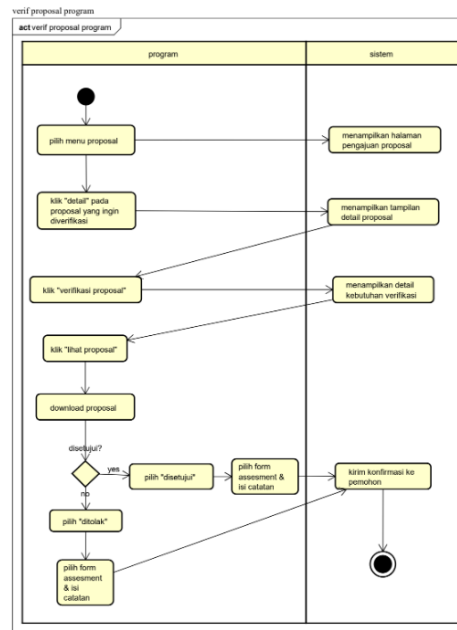
Gambar 4. 16 Activity diagram bagian verifikasi proposal manager

Activity diagram bagian badan pengurus dilakukan oleh *user* badan pengurus, dimana badan pengurus akan melakukan verifikasi apakah proposal yang diberikan akan disetujui ataupun ditolak. Berikut activity diagram dapat dilihat pada gambar 4.17.



Gambar 4. 17 Activity diagram bagian verifikasi proposal badan pengurus

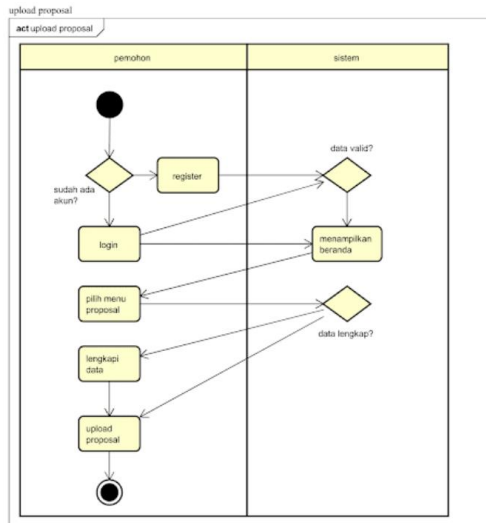
Activity diagram bagian program dilakukan oleh *user* program, dimana program akan melakukan verifikasi proposal dengan mengirimkan file assesment, serta memberitahukan apakah proposal yang diajukan disetujui oleh program atau tidak. Berikut activity diagram dapat dilihat pada gambar 4.18.



Gambar 4. 18 Activity diagram bagian program

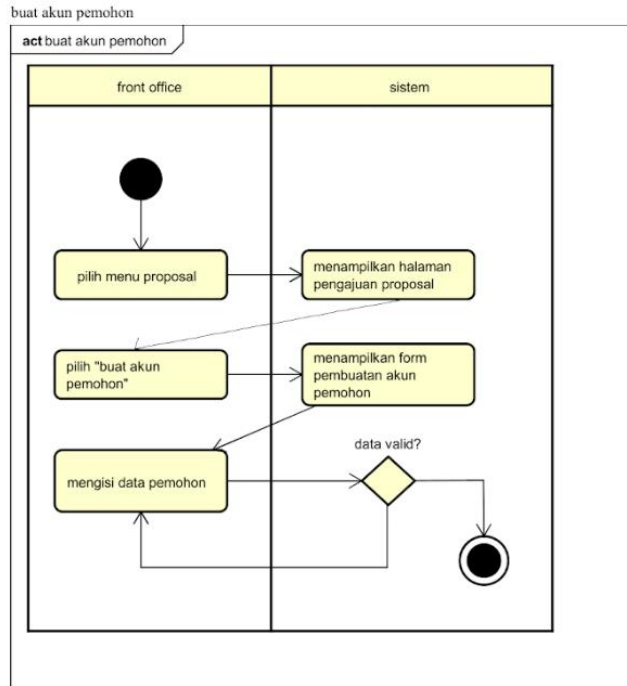
Activity diagram bagian upload proposal front-office dilakukan oleh front-office, dimana front-office dapat menambahkan proposal jika memang terdapat case pemohon yang datang langsung dan ingin ditangani secara offline. Berikut activity diagram dapat dilihat pada gambar 4.19.





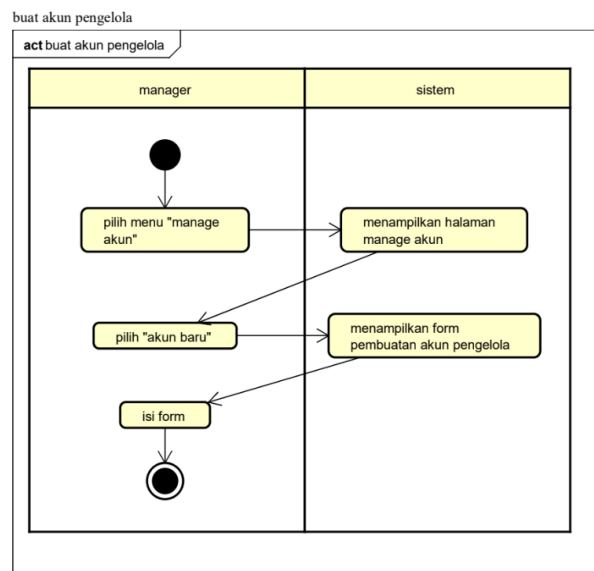
Gambar 4. 19 Activity diagram bagian upload proposal

Activity diagram bagian register pemohon dilakukan oleh pemohon, dimana pemohon yang belum memiliki akun dan ingin membuat akun masuk ke menu buat akun yang ada di halaman login, lalu mengisikan data yang tertera di form, dan nantinya setelah data dikirim mereka akan mendapatkan notifikasi email untuk melakukan verifikasi akun mereka, sebelum akun mereka siap untuk digunakan. Berikut activity diagram dapat dilihat pada gambar 4.20.



Gambar 4.20 Activity diagram buat akun pemohon

Activity diagram bagian buat akun pengelola dilakukan oleh *user manager*, dimana *manager* dapat menambahkan pengelola pada kantor mereka (front-office / badan pengurus) sesuai dengan kebutuhan sang *manager*. Berikut activity diagram dapat dilihat pada gambar 4.21.



Gambar 4.21 Activity diagram buat akun pengelola

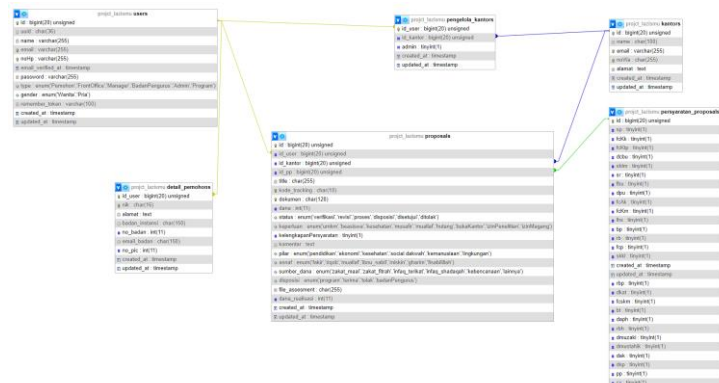
### 3) Implementasi Coding dan Hasil Implementasi

#### a) Setting database (Migrations & Seeders)

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->uuid();
17             $table->string('name');
18             $table->string('email')->unique();
19             $table->string('noHp')->unique();
20             $table->timestamp('email_verified_at')->nullable();
21             $table->string('password');
22             $table->enum('type', ['Pemohon', 'FrontOffice', 'Manager', 'BadanPengurus', 'Admin',
23                                     'Program'])->default('Pemohon'); //add type boolean Users:
24
25             @=>User, 1=>Admin, 2=>Manager
26             $table->enum('gender', ['Wanita', 'Pria']);
27             $table->rememberToken();
28             $table->timestamps();
29         });
30     }
31
32     /**
33      * Reverse the migrations.
34      */
35     public function down(): void
36     {
37         Schema::dropIfExists('users');
38     }
39 };
```

Gambar 4.22 Listing code setting database

Listing code pada gambar 4.22 berguna untuk membuat table database yang sebelumnya sudah dirancang sesuai dengan kebutuhan dan permintaan user, untuk dibuat versi code-nya supaya nanti dapat otomatis di migrate dengan isian yang sudah ditulis. Dapat dilihat pada gambar 4.23 merupakan hasil dari migrations yang sudah di migrate.



Gambar 4.23 Hasil migrations

```

1 <?php
2
3 namespace Database\Seeders;
4
5 use App\Models\User;
6 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
7 use Illuminate\Database\Seeder;
8 use Illuminate\Support\Facades\Hash;
9 use Illuminate\Support\Str;
10
11 class UserSeeder extends Seeder
12 {
13     /**
14      * Run the database seeds.
15      */
16     public function run(): void
17     {
18         $data = [
19             [
20                 'name' => 'admin',
21                 'uid' => Str::uuid(),
22                 'email' => 'admin@bs.local',
23                 'nohp' => '002989182921',
24                 'email_verified_at' => '2023-11-27 11:31:26',
25                 'password' => Hash::make('123'),
26                 'type' => 'Admin',
27             ],
28             [
29                 'name' => 'badan pengurus',
30                 'uid' => Str::uuid(),
31                 'email' => 'badan@bs.local',
32                 'nohp' => '002989182922',
33                 'email_verified_at' => '2023-11-27 11:31:26',
34                 'password' => Hash::make('123'),
35                 'type' => 'BadanPengurus',
36             ],
37             [
38                 'name' => 'Program',
39                 'uid' => Str::uuid(),
40                 'email' => 'program@bs.local',
41                 'nohp' => '002989182901',
42                 'email_verified_at' => '2023-11-27 11:31:26',
43                 'password' => Hash::make('123'),
44                 'type' => 'Program',
45             ],
46             [
47                 'name' => 'manager',
48                 'uid' => Str::uuid(),
49                 'email' => 'manager@bs.local',
50                 'nohp' => '002989182923',
51                 'email_verified_at' => '2023-11-27 11:31:26',
52                 'password' => Hash::make('123'),
53                 'type' => 'Manager',
54             ],
55             [
56                 'name' => 'front office',
57                 'uid' => Str::uuid(),
58                 'email' => 'front@bs.local',
59                 'nohp' => '002989182924',
60                 'email_verified_at' => '2023-11-27 11:31:26',
61                 'password' => Hash::make('123'),
62                 'type' => 'FrontOffice',
63             ],
64             [
65                 'name' => 'pemohon',
66                 'uid' => Str::uuid(),
67                 'email' => 'pemohon@bs.local',
68                 'nohp' => '002989182925',
69                 'email_verified_at' => '2023-11-27 11:31:26',
70                 'password' => Hash::make('123'),
71                 'type' => 'Pemohon',
72             ],
73         ];
74         foreach ($data as $k => $v) {
75             User::create($v);
76         }
77     }
78 }
79
80

```

Gambar 4.24 Listing code setting seeders

Listing code pada gambar 4.24 berguna untuk membuat data awal saat aplikasi dijalankan, supaya nantinya saat mencoba fitur dalam aplikasi front-end ataupun anggota team yang lain ingin melakukan testing terhadap aplikasinya tidak kesusahan. Dapat dilihat pada gambar 4.25 merupakan hasil dari seeders ketika dijalankan.

| id | user        | name         | email                 | verified | created_at          | password                             | role         | gender | membership_status | country_id | updated_at          |
|----|-------------|--------------|-----------------------|----------|---------------------|--------------------------------------|--------------|--------|-------------------|------------|---------------------|
| 1  | admin       | Admin        | admin@telem.com       | 1        | 2024-11-20 10:15:00 | \$2y\$12\$5t1u6dKfPa.80Ppa.r0k4u6L0C | Admin        | Male   | 1                 | 1          | 2024-11-20 10:15:00 |
| 2  | program     | Program      | program@telem.com     | 1        | 2024-11-20 10:15:00 | \$2y\$12\$5t1u6dKfPa.80Ppa.r0k4u6L0C | Program      | Male   | 1                 | 1          | 2024-11-20 10:15:00 |
| 3  | manager     | Manager      | manager@telem.com     | 1        | 2024-11-20 10:15:00 | \$2y\$12\$5t1u6dKfPa.80Ppa.r0k4u6L0C | Manager      | Male   | 1                 | 1          | 2024-11-20 10:15:00 |
| 4  | frontoffice | Front Office | frontoffice@telem.com | 1        | 2024-11-20 10:15:00 | \$2y\$12\$5t1u6dKfPa.80Ppa.r0k4u6L0C | Front Office | Male   | 1                 | 1          | 2024-11-20 10:15:00 |
| 5  | member      | Member       | member@telem.com      | 1        | 2024-11-20 10:15:00 | \$2y\$12\$5t1u6dKfPa.80Ppa.r0k4u6L0C | Member       | Male   | 1                 | 1          | 2024-11-20 10:15:00 |

Gambar 4. 25 Hasil setelah seeders dijalankan

### b) Setting Model

```

1 <?php
2 namespace App\Models;
3
4 use Illuminate\Contracts\Auth\MustVerifyEmail;
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7 use Illuminate\Notifications\Notifiable;
8 use Laravel\Sanctum\HasApiTokens;
9
10 class User extends Authenticatable implements MustVerifyEmail
11 {
12     use HasApiTokens, HasFactory, Notifiable;
13
14     /**
15      * The attributes that are mass assignable.
16      */
17     protected $fillable = [
18         'name',
19         'email',
20         'password',
21     ];
22
23     /**
24      * The attributes that should be hidden for serialization.
25      */
26     protected $hidden = [
27         'password',
28         'remember_token',
29     ];
30
31     /**
32      * The attributes that should be hidden for serialization.
33      */
34     protected $casts = [
35         'password' => 'string',
36         'remember_token' => 'string',
37     ];
38
39     /**
40      * The attributes that should be cast to native types.
41      */
42     protected $casts = [
43         'password' => 'string',
44     ];
45
46     /**
47      * The attributes that should be cast to native types.
48      */
49     protected $casts = [
50         'password' => 'string',
51         'remember_token' => 'string',
52     ];
53
54     /**
55      * The attributes that should be cast to native types.
56      */
57     protected $casts = [
58         'password' => 'string',
59     ];
60
61     /**
62      * The attributes that should be cast to native types.
63      */
64     protected $casts = [
65         'password' => 'string',
66     ];
67
68     /**
69      * The attributes that should be cast to native types.
70      */
71     protected $casts = [
72         'password' => 'string',
73     ];
74
75     /**
76      * The attributes that should be cast to native types.
77      */
78     protected $casts = [
79         'password' => 'string',
80     ];
81
82     /**
83      * The attributes that should be cast to native types.
84      */
85     protected $casts = [
86         'password' => 'string',
87     ];
88
89     /**
90      * The attributes that should be cast to native types.
91      */
92     protected $casts = [
93         'password' => 'string',
94     ];
95
96     /**
97      * The attributes that should be cast to native types.
98      */
99     protected $casts = [
100         'password' => 'string',
101     ];
102
103     /**
104      * The attributes that should be cast to native types.
105      */
106     protected $casts = [
107         'password' => 'string',
108     ];
109
110     /**
111      * The attributes that should be cast to native types.
112      */
113     protected $casts = [
114         'password' => 'string',
115     ];
116
117     /**
118      * The attributes that should be cast to native types.
119      */
120     protected $casts = [
121         'password' => 'string',
122     ];
123
124     /**
125      * The attributes that should be cast to native types.
126      */
127     protected $casts = [
128         'password' => 'string',
129     ];
130
131     /**
132      * The attributes that should be cast to native types.
133      */
134     protected $casts = [
135         'password' => 'string',
136     ];
137
138     /**
139      * The attributes that should be cast to native types.
140      */
141     protected $casts = [
142         'password' => 'string',
143     ];
144
145     /**
146      * The attributes that should be cast to native types.
147      */
148     protected $casts = [
149         'password' => 'string',
150     ];
151

```

Gambar 4. 26 Listing code setting model

Listing code pada gambar 4.26 berguna untuk melakukan pengaturan data apa saja yang dapat di-isi oleh user serta data apa

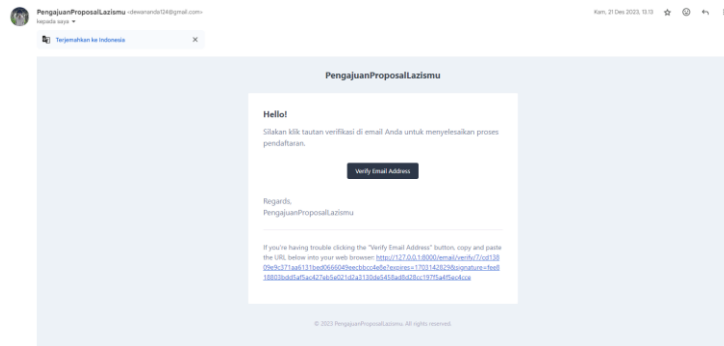
yang harus ditutupi demi keamanan seperti data password, token dsb. Model juga berfungsi untuk mempermudah dalam pembuatan fitur di controller nantinya.

### c) Setting mail

```
1 <?php
2
3 namespace App\Mail;
4
5 use Illuminate\Bus\Queueable;
6 use Illuminate\Contracts\Queue\ShouldQueue;
7 use Illuminate\Mail\Mailable;
8 use Illuminate\Mail\Mailables\Content;
9 use Illuminate\Mail\Mailables\Envelope;
10 use Illuminate\Queue\SerializesModels;
11
12 class VerificationAccountMail extends Mailable
13 {
14     use Queueable, SerializesModels;
15
16     public $user;
17     /**
18      * Create a new message instance.
19      */
20     public function __construct($user)
21     {
22         //
23         $this->user = $user;
24     }
25
26     /**
27      * Get the message envelope.
28      */
29     public function envelope(): Envelope
30     {
31         return new Envelope(
32             subject: 'Verification Account Mail',
33         );
34     }
35
36     /**
37      * Get the message content definition.
38      */
39     public function content(): Content
40     {
41         return new Content(
42             markdown: 'emails.verifikasiEmailPassword',
43             with: [
44                 'user' => $this->user,
45             ]
46         );
47     }
48
49     /**
50      * Get the attachments for the message.
51      *
52      * @return array<int, \Illuminate\Mail\Mailables\Attachment>
53      */
54     public function attachments(): array
55     {
56         return [];
57     }
58 }
59
```

Gambar 4. 27 Listing code setting mail

*Listing code* pada gambar 4.27 berguna untuk melakukan konfigurasi terhadap response email yang dimana saat *user* melakukan tindakan untuk meminta sand email maka secara otomatis aplikasi akan mengirimkan email ke alamat tujuan. Dapat dilihat pada gambar 4. merupakan hasil ketika *user* menerima email dari aplikasi.



Gambar 4. 28 Hasil ketika user menerima email dari aplikasi

#### d) Setting Route

```
1 <?php
2
3 /
4
5 -----
6 Web Routes
7 -----
8
9 Here is where you can register web routes for your application. These
10 routes are loaded by the RouteServiceProvider and all of them will
11 be assigned to the "web" middleware group. Make something great!
12 /
13
14 // testing
15 Route::get('/view-mail', function(){
16     $proposal = proposal::get()->first();
17     $user = User::get()->first();
18     $kantor = User::get()->first();
19
20     $mail = new App\Mail\updateVerifikasIProposalMail($proposal, $user, $kantor);
21     return $mail->render();
22 });
23
24 Route::get('/', [AuthController::class, 'index']);
25 Route::post('/track', [AuthController::class, 'getTrackBerkas'])->name('trackingBerkas');
26
27 Route::get('/email/verify', function () {
28     return view('auth.verify-email');
29 })->middleware('auth')->name('verification.notice');
30
31 Route::get('/email/verify/{id}/{hash}', function (EmailVerificationRequest $request) {
32     $request->fulfill();
33
34     return redirect(route('login'))->with('message', 'Akun anda berhasil diverifikasi! silahkan login');
35 })->middleware('auth', 'signed')->name('verification.verify');
36
37 Route::post('/email/verification-notification', function (Request $request) {
38     $request->user()->sendEmailVerificationNotification();
39
40     return redirect(route('login'))->with('success', 'Email untuk verification sudah dikirim, silahkan cek email
41     anda!');
42 })->middleware('auth', 'throttle:6,1')->name('verification.send');
43
44 Route::controller(AuthController::class)->group(function () {
45     Route::get('register', 'register')->name('register');
46     Route::post('register', 'registerSave')->name('register.save');
47     Route::post('login', 'loginAction')->name('login.action');
48     Route::get('login', 'login')->name('login');
49     Route::get('logout', 'logout')->middleware('auth')->name('logout');
50
51     Route::get('viewForgetPassword', 'viewForgetPassword')->nam
```

Gambar 4. 29 Listing code setting route

Listing code pada gambar 4.29 berguna untuk menentukan URL yang akan dipetakan ke controller tertentu dalam aplikasi *Laravel*. Pada kode diatas juga sudah ditambahkan middleware guna untuk memberikan pengamanan terhadap beberapa URL, dimana middleware dapat digunakan untuk berbagai macam tujuan, seperti autentikasi, otorisasi, validasi, dan sebagainya.hanya user dengan hak akses sesuai yang dapat membuka url tersebut.

#### e) Fitur Tracking Proposal

```
1 public function getTrackBerkas(Request $request)
2 {
3     $data = $request->input('data');
4     $post = User::where('id', $data)->get()->first();
5
6     $data = [
7         "data" => $post,
8     ];
9
10    if ($post == null) {
11        echo '<script>alert("Data tidak ada");</script>';
12        return redirect("/");
13    } else {
14        return view("proposal.detailProposal", ['data' => $post]);
15    }
16 } res ? go(f, res[1], acc.concat([res[0]])) : acc
17 }
18 return go(f, seed, [])
19 }
```

Gambar 4. 30 Listing code fitur tracking proposal

Listing code pada gambar 4. merupakan fungsi dari code yang dibuat untuk membaca ke *database* di *table* proposal bagian *kode\_tracking*, apakah *kode\_tracking* tersebut tersedia atau tidak. Jika tersedia maka akan menampilkan status proposal dengan isian data sesuai yang dibutuhkan oleh *front-end*.

#### f) Fitur Login

```
1 public function login()
2 {
3     return view('auth/login');
4 }
5
6 public function loginAction(Request $request)
7 {
8     Validator::make($request->all(),
9     [
10        'email' => 'required|email',
11        'password' => 'required',
12    ], [
13        // 'confirmed' => 'Password yang anda kintakan kurang tepat',
14        'required' => 'Kolom :attribute tidak boleh kosong'
15    ]->validate();
16
17    if (!Auth::attempt($request->only('email', 'password'), $request->boolean('remember'))) {
18        throw ValidationException::withMessages([
19            'email' => trans('auth.failed')
20        ]);
21        Log::create([
22            'users_id' => auth()->user()->id,
23            'name' => auth()->user()->name,
24            'email' => auth()->user()->email,
25            'noHp' => auth()->user()->noHp,
26            'ip' => $_SERVER['REMOTE_ADDR'],
27            'job' => 'loginfailed',
28        ]);
29    }
30
31    Log::create([
32        'users_id' => auth()->user()->id,
33        'name' => auth()->user()->name,
34        'email' => auth()->user()->email,
35        'noHp' => auth()->user()->noHp,
36        'ip' => $_SERVER['REMOTE_ADDR'],
37        'job' => 'loginsuccess',
38    ]);
39
40    $request->session()->regenerate();
41
42    return redirect()->route('dashboard')->with('success', 'You have successfully login.');
```

Gambar 4. 31 Listing code fitur login



*Listing code* pada gambar 4.31 merupakan fungsi untuk menampilkan halaman login (login), dan juga untuk memproses login (loginAction). Dimana untuk loginAction terdapat fungsi untuk mengecek apakah data yang dikirimkan oleh *user* valid atau tidak dengan menggunakan *validator* dari fungsi bawaan *laravel*, lalu jika valid dan sesuai dengan yang *didatabase* maka *user* akan dibawa untuk masuk ke halaman dashboard sesuai role mereka masing masing.

g) Fitur Logout

```

1 public function logout(Request $request)
2 {
3     Log::create([
4         'users_id' => auth()->user()->id,
5         'name' => auth()->user()->name,
6         'email' => auth()->user()->email,
7         'noHp' => auth()->user()->noHp,
8         'ip' => $_SERVER['REMOTE_ADDR'],
9         'job' => 'logout',
10    ]);
11
12    Auth::guard('web')->logout();
13
14    $request->session()->invalidate();
15
16    return redirect(route('login'))->with('success', 'You have successfully logout.');
```

Gambar 4.32 *Listing code* fitur logout

*Listing code* pada gambar 4.32 merupakan fungsi untuk melakukan logout dari aplikasi, dimana fungsi ini akan membuat session yang sebelumnya dibuat saat fungsi login, dirubah menjadi

h) Fitur Buat Akun

```

1 public function registerSave(Request $request)
2 {
3     Validator::make($request->all(), [
4         'name' => 'required',
5         'email' => 'required|email|unique:users,email',
6         'noHp' => 'required|min:11|unique:users,noHp',
7         'gender' => 'required',
8         'password' => ['required', 'confirmed', 'Password::min(6)->mixedCase()'],
9     ], [
10        'required' => 'Kolom :attribute tidak boleh kosong',
11        'min' => 'Kolom :attribute harus memiliki minimal :min karakter',
12        'mixed' => 'Kolom :attribute harus memiliki minimal satu huruf kapital dan kecil',
13        'confirmed' => 'Kolom :attribute confirmation tidak cocok',
14        'unique' => ':attribute sudah digunakan',
15        'email' => 'kolom :attribute harus disisakan dengan email yang valid'
16    ])->validate();
17
18    $user = User::create([
19        'name' => $request->name,
20        'uid' => Str::uuid($request->name),
21        'email' => $request->email,
22        'noHp' => $request->noHp,
23        'password' => Hash::make($request->password),
24        'gender' => $request->gender,
25        'type' => 'Pemohon',
26    ]);
27
28    detail_pemohon::create([
29        'id_user' => $user->id
30    ]);
31
32    event(new Registered($user));
33
34    Auth::login($user);
35
36    return redirect()->route('login')->with('success', 'Your account successfully created');
```

Gambar 4. 33 *Listing code* fitur buat akun

*Listing code* pada gambar 4.33 merupakan fungsi untuk membuat akun pemohon, yang dimana terdapat fitur *validator*

untuk mengecek apakah inputan dari *user* sudah sesuai, dan jika tidak sesuai maka akan memunculkan error. Lalu jika inputan *user* sesuai semua maka akan dibuatkan akun serta diberikan email untuk melakukan verifikasi email dari aplikasi, supaya dapat login.

#### i) Verifikasi Akun

```
1 public function verificationEmail($token, Request $request)
2 {
3     Validator::make($request->all(), [
4         'password' => ['required', 'confirmed', 'passwords:mn(8)-mixedCase()],
5         'password_confirmation' => 'required'
6     ]), [
7         'required' => 'Kolom :attribute tidak boleh kosong',
8         'min' => 'Kolom :attribute harus memiliki minimal :min karakter',
9         'mixed' => 'Kolom :attribute harus memiliki minimal satu huruf kapital dan kecil',
10        'confirmed' => 'kolom :attribute confirmation tidak cocok',
11    ]->validate();
12
13    $user = User::where('remember_token', '=', $token)->get()->first();
14
15    if (! empty($user) )
16    {
17        $user->password = Hash::make($request->password);
18        $user->remember_token = null;
19        $user->email_verified_at = date('Y-m-d h:i:s');
20        $user->save();
21
22        return redirect('login')->with('success', 'Anda berhasil melakukan verifikasi, silahkan login!');
23    }
24    else
25    {
26        abort(404);
27    }
28 }
```

Gambar 4. 34 Listing code verifikasi akun yang dibuatkan oleh pengelola

*Listing code* pada gambar 4.34 merupakan fungsi untuk melakukan verifikasi akun ketika *user* dibuatkan akunnya oleh pengelola kantor, yang dimana nantinya setelah *user* mendapatkan email dari aplikasi, nantinya akan diarahkan ke halaman web untuk membuat password sesuai keinginan mereka, dan setelah password dibuat maka akun mereka akan otomatis terbuat.

```
1 Route::get('/email/verify/{id}/{hash}', function (EmailVerificationRequest $request) {
2     $request->fulfill();
3
4     return redirect(route('login'))->with('message', 'Akun anda berhasil diverifikasi! silahkan login');
5 }->middleware(['auth', 'signed'])->name('verification.verify');
```

Gambar 4. 35 Listing code verifikasi akun yang dibuat sendiri oleh *user*

*Listing code* pada gambar 4.35 merupakan fungsi untuk melakukan verifikasi akun ketika *user* melakukan pendaftaran sendiri (pemohon), yang dimana nantinya mereka akan mendapatkan email dari aplikasi, dan mereka hanya perlu melakukan klik pada tombol yang ada di halaman web mereka buka sebelumnya. Ini merupakan fungsi bawaan dari laravel yang memiliki nama “EmailVerificationRequest”.

## j) Lupa Password

```
1 public function sandforgotPassword(Request $request)
2 {
3     $user = User::where('email', '=', $request->email)->get()->first();
4
5     if(! empty($user))
6     {
7         $user->remember_token = Str::random(40);
8         $user->save();
9
10        Mail::to($user->email)->send(new ForgotPasswordMail($user));
11
12        return redirect()->back()->with('success', 'Silahkan cek email anda!');
13    }else{
14        return redirect()->back()->with('error', 'Email yang anda berikan belum terdaftar!');
15    }
16 }
```

Gambar 4. 36 Listing code untuk melakukan permintaan lupa password

Listing code pada gambar 4.36 merupakan fungsi untuk melakukan permintaan untuk diberikan email lupa password dari aplikasi.

```
1 public function resetPasswordWT($token, Request $request)
2 {
3     Validator::make($request->all(), [
4         'password' => ['required', 'confirmed', Password::min(8)->mixedCase()]
5     ], [
6         'required' => 'Kolom :attribute tidak boleh kosong',
7         'min' => 'Kolom :attribute harus memiliki minimal :min karakter',
8         'mixed' => 'Kolom :attribute harus memiliki minimal satu huruf kapital dan kecil',
9         'confirmed' => 'kolom :attribute confirmation tidak cocok',
10    ])->validate();
11
12    $user = User::where('remember_token', '=', $token)->get()->first();
13
14    if( ! empty($user) )
15    {
16        $user->password = Hash::make($request->password);
17        $user->remember_token = null;
18        $user->save();
19
20        return redirect('login')->with('success', 'Anda berhasil memperbarui password!');
21    }else
22    {
23        abort(404);
24    }
25 }
```

Gambar 4. 37 Listing code untuk menerapkan fungsi ganti password

Listing code pada gambar 4.37 merupakan fungsi untuk melakukan ganti password, dimana pada fungsi ini pertama diberikan *validator* pada form-nya, lalu jika isian form yang diberikan oleh *user* tidak ada masalah maka akan dilanjutkan untuk melakukan *update* password sesuai dengan yang dibuat oleh *user* sebelumnya.

## k) Fitur Notifikasi Email

```
1 class VerificationAccountMail extends Mailable
2 {
3     use Queueable, SerializesModels;
4     public $user;
5     /**
6      * Create a new message instance.
7      */
8     public function __construct($user)
9     {
10         //
11         $this->user = $user;
12     }
13     /**
14      * Get the message envelope.
15      */
16     public function envelope(): Envelope
17     {
18         return new Envelope(
19             subject: 'Verification Account Mail',
20         );
21     }
22     /**
23      * Get the message content definition.
24      */
25     public function content(): Content
26     {
27         return new Content(
28             markdown: 'emails.verifikasiEmailPassword',
29             with: [
30                 'user' => $this->user,
31             ]
32         );
33     }
34     /**
35      * Get the attachments for the message.
36      * @return array<int, Illuminate\Mail\Mailables\Attachment>
37      */
38     public function attachments(): array
39     {
40         return [];
41     }
42 }
43 }
44 }
45 }
46 }
47 }
```

Gambar 4. 38 Listing code untuk memberikan notifikasi email dari aplikasi

Listing code pada gambar 4.38 merupakan fungsi untuk memberikan notifikasi email kepada *user* yang dituju sesuai dengan penerapan pada controller, karena nantinya controller akan memanggil fungsi mail ini pada fungsi mereka, dan akan otomatis mengirimkan email yang sesuai dengan isian pada fungsi content di line 28, dengan memberikan data *user*.

## I) Fitur Upload Proposal

```
1 public function tambahProposal(Request $request)
2 {
3     $kode_tracking = '';
4     $depan = 'JOG';
5     $characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
6     $length = 5; // Panjang kode tracking yang diinginkan
7     for ($i = 0; $i < $length; $i++) {
8         $kode_tracking .= $characters[rand(0, strlen($characters) - 1)];
9     }
10
11     $kode_tracking = $depan . $kode_tracking;
12
13     $syaratProp = persyaratan_proposal::create();
14     $syaratProp->changeBaseType($request->jenis_proposal);
15
16
17     $data = new proposal();
18     $data->id_user = auth()->user()->id;
19     $data->id_kantor = $request->target_kantor;
20     $data->id_pp = $syaratProp->id;
21     $data->skeperluan = $request->jenis_proposal;
22     $data->title = $request->judul;
23     $data->kode_tracking = $kode_tracking;
24     $data->status = 'verifikasi';
25     $data->dana = $request->dana;
26
27     (($request->target_kantor == 'none')
28     {
29         return redirect('/data-proposal/TambahProposal')->with('error', 'Pemohon harus memilih lokasi tempat
30 pengajuan');
31     }
32     if ($request->file('file'))
33     {
34         $dokumen = $request->file('file');
35         $mime = mime_content_type($dokumen->getPathname());
36
37         if ($mime == 'application/pdf') {
38             // $dokumen->getRealPath();
39             if ($dokumen->getSize() >= 2097152) {
40                 return redirect('/data-proposal/TambahProposal')->with('error', 'Ukuran file maksimum yang
41 diperbolehkan adalah 2MB');
42             }
43             $namaDokumen = 'Prop' . date('Yndhis') . '.' . $dokumen->getClientOriginalExtension();
44             Storage::disk('public')->putFileAs('dokumen/' . $kode_tracking , new File($dokumen), $namaDokumen);
45             $data->dokumen = $namaDokumen;
46             $data->save();
47
48             return redirect('/data-proposal')->with('success', 'Proposal berhasil di ajukan, silahkan tunggu beberapa
49 hari untuk proses selanjutnya!');
50         } else {
51             // file bukan PDF, handle kesalahan di sini
52             return redirect('/data-proposal/TambahProposal')->with('error', 'File harus berformat PDF');
53         }
54     }
55 }
56 }
```

Gambar 4. 39 Listing code untuk upload proposal

*Listing code* pada gambar 4.39 merupakan fungsi untuk melakukan upload proposal, yang dimana nantinya juga dibuatkan otomatis kode trackingnya, lalu juga ada handler supaya *user* melakukan upload proposal sesuai dengan yang diharuskan. Dan jika *user* telah melakukan upload proposal sesuai dengan ketentuan nantinya proposal akan ditambahkan di *database* dengan status “verifikasi” yang dimana pihak front office akan melakukan pengecekan terhadap proposal tersebut.

## m) Fitur Verifikasi Proposal

```
1 public function updateProposal(Request $request, $id)
2 {
3     $type = auth()-user()->typeker();
4     $proposal = Proposal::findOrFail($id);
5     $user = User::findOrFail($proposal->id_user);
6     $kantor = Kantor::findOrFail($proposal->id_kantor);
7
8     if($proposal == null)
9     {
10        return redirect(route('proposal.'. $type));
11    }
12    else
13    {
14        out($type){
15            case('o'):
16                switch ($proposal->keperluan) {
17                    case 'umum':
18                        persyaratan_proposal::find($proposal->id_pp)->updateSingleState('sp', $request->sp);
19                        persyaratan_proposal::find($proposal->id_pp)->updateSingleState('dcbu', $request->dcbu);
20                        persyaratan_proposal::find($proposal->id_pp)->updateSingleState('cktp', $request->cktp);
21                        persyaratan_proposal::find($proposal->id_pp)->updateSingleState('ckk', $request->ckk);
22                        persyaratan_proposal::find($proposal->id_pp)->updateSingleState('skln', $request->skln);
23                        persyaratan_proposal::find($proposal->id_pp)->updateSingleState('sr', $request->sr);
24                        persyaratan_proposal::find($proposal->id_pp)->updateSingleState('fhu', $request->fhu);
25                        persyaratan_proposal::find($proposal->id_pp)->updateSingleState('dpu', $request->dpu);
26                    }
27                }
28
29                if ($persyaratan_proposal::find($proposal->id_pp)->checkVerificationProposal($proposal->keperluan) )
30                {
31                    $proposal->status = 'proses';
32                    $proposal->save();
33                    return redirect(route('proposal.'. $type))->with('success_next_step_for', 'Proposal berhasil
34dijalankan ketahap selanjutnya, karena seluruh persyaratan telah lengkap');
35                }
36                $proposal->status = 'verifikasi';
37                $proposal->save();
38
39                return redirect(route('proposal.'. $type))->with('success', 'Proposal berhasil diupdate, dan masih
40terdapat beberapa kekurangan');
41            }
42            case('m'):
43                switch ($request->status){
44                    case('dijetujui'):
45                        if($request->asnaf == 'null' || $request->asnaf == 'null' ||
46                            $request->sumber_dana == 'null' || $request->realisasi_dana == null)
47                        {
48                            return redirect()->route('ViewVerifikasiProposal.m', $id)->with('error', 'Kolom pilar,
49asnaf, sumber dana, dan realisasi dana wajib di isi untuk melengkapi prasyarajan');
50                        }
51                        else
52                        {
53                            $proposal->update([
54                                'pilar' => $request->pilar,
55                                'sumber_dana' => $request->sumber_dana,
56                                'dana_realisasi' => $request->realisasi_dana,
57                                'asnaf' => $request->asnaf,
58                                'disposisi' => null,
59                                'status' => 'dijetujui'
60                            ]);
61                            break;
62                        }
63                    case('ditolak'):
64                        $proposal->status = $request->status;
65                        break;
66                    case('disposisi'):
67                        if($request->dlapo == 'null')
68                        {
69                            return redirect()->route('ViewVerifikasiProposal.m', $id)->with('error', 'Kolom
70dlapo wajib di isi untuk melengkapi form disposisi');
71                        }
72                        else
73                        {
74                            $proposal->update([
75                                'status' => $request->status,
76                                'disposisi' => $request->dlapo,
77                            ]);
78                            break;
79                        }
80                    case('revisi'):
81                        if($request->catatan == null)
82                        {
83                            return redirect()->route('ViewVerifikasiProposal.m', $id)->with('error', 'Kolom
84catatan wajib di isi untuk melengkapi form revisi');
85                        }
86                        else
87                        {
88                            $proposal->update([
89                                'status' => 'revisi',
90                                'komentar' => $request->catatan,
91                            ]);
92                            break;
93                        }
94                        default:
95                            return redirect()->route('ViewVerifikasiProposal.m', $id)->with('error', 'Anda harus
96memilih status untuk proposal ini');
97                            break;
98                        }
99                    case('bp'):
100                        switch ($request->status){
101                            case('dijetujui'):
102                                $proposal->status = $request->status;
103                                break;
104                            case('ditolak'):
105                                $proposal->status = $request->status;
106                                break;
107                        }
108                        $proposal->asnaf = $request->asnaf;
109                        $proposal->sumber_dana = $request->dana;
110                        $proposal->save();
111                        break;
112                    case('pr'):
113                        if($request->status == null)
114                        {
115                            return redirect()->back()->with('error', 'Anda harus menentukan status dari proposal');
116                        }
117                        if($request->file('file_assesment'))
118                        {
119                            if($proposal->file_asesment != null)
120                            {
121                                Storage::disk('public')->delete('dokumen/'. $proposal->kode_tracking . '/' . $proposal->
122file_asesment);
123                            }
124                            $dokumen = $request->file('file_assesment');
125                            $mime = mime_content_type($dokumen->getPathname());
126                            if ($mime == 'application/pdf')
127                            {
128                                $namaDokumen = 'Assesment_' . date('Yndhis') . '_' . $dokumen->getClientOriginalExtension();
129                                Storage::disk('public')->putFileAs('dokumen/'. $proposal->kode_tracking , new
130File($dokumen), $namaDokumen);
131                                $proposal->update([
132                                    'file_asesment' => $namaDokumen,
133                                    'status' => 'proses',
134                                    'disposisi' => $request->status,
135                                    'komentar' => $request->catatan
136                                ]);
137                                break;
138                            }
139                            else
140                            {
141                                // file bukan PDF, handle ke halaman di atas
142                                return redirect('/data-proposal/TambahProposal')->with('error', 'File harus berbentuk
143PDF');
144                            }
145                        }
146                    }
147                }
148                }
149            }
150        }
151    }
152    // Mail::to($user->email)->send(new updateVerifikasiProposalMail($proposal, $user, $kantor));
153    return redirect(route('proposal.'. $type))->with('success', 'Proposal berhasil diverifikasi');
154 }
155 }
```

Gambar 4. 40 Listing code untuk fitur verifikasi proposal

Listing code pada gambar 4.40 merupakan fungsi untuk melakukan verifikasi proposal yang dimana setiap level pengelola

dari front office, manager, program, badan pengurus memiliki caranya sendiri untuk melakukan verifikasi proposal, seperti contohnya jika *user* front office melakukan verifikasi proposal maka front office haruslah melakukan *update* pada data yang ada di persyaratan proposal yang sebelumnya 0 menjadi 1, dimana jika 0 berarti persyaratan proposal tersebut belum terpenuhi dan jika 1 berarti sudah terpenuhi. Untuk manager mereka perlu melakukan verifikasi proposal dengan menentukan apakah proposal tersebut disetujui, ditolak, disposisi, ataupun direvisi. Para pengelola juga dapat melakukan komentar jika memang diperlukan.

#### n) Fitur Riwayat Proposal

```

1 public function viewHistoryProposal()
2 {
3     $dataUser = Auth::user();
4     $data = [
5         'datatables' => Proposal::join('kantors', 'proposals.id_kantor', '=', 'kantors.id')
6         ->select('proposals.*', 'kantors.name', 'kantors.alamat')
7         ->orderBy('updated_at', 'asc')
8         ->where('status', 'Ditolak')
9         ->where('status', 'Disetujui')
10        ->where('id_user', $dataUser->id)
11        ->get(),
12
13        'user' => $dataUser,
14        'addProposal' => 0,
15    ];
16 }
17 return view('pemohon.index', $data);
18 }
19 }
20

```

Gambar 4. 41 Listing code fungsi riwayat proposal

*Listing code* pada gambar 4.41 merupakan fungsi untuk menampilkan riwayat proposal yang sesuai dengan role *user* dan juga hanya diambil data proposal yang memiliki status disetujui ataupun ditolak.

#### o) Fitur CRUD Akun Pengelola Melalui *Manager*

```

1 public function viewDataPengelola()
2 {
3     $user = auth::user();
4     $data_pengelola = User::join('pengelola_kantors', 'users.id', 'pengelola_kantors.id_user')->get();
5     $user = $data_pengelola->find($user->id);
6     $data_pengelola = $data_pengelola->where('id_kantor', $user->id_kantor);
7
8     $data = [
9         'datatables' => $data_pengelola->whereIn('type', ['FrontOffice', 'BadanPengurus', 'Program']),
10    ];
11
12
13
14    return view('pengelola.DataPengelola.index', $data);
15 }

```

Gambar 4. 42 Listing code read data pengelola

*Listing code* pada gambar 4.42 merupakan fungsi untuk mengambil data pengelola sesuai dengan role dan tempat kantor.

```

1 public function TambahAkunPengelola(Request $request)
2 {
3     Validator::make($request->all(), [
4         'name' => 'required',
5         'noHp' => 'required|unique:users',
6         'email' => 'required|email|unique:users',
7         'gender' => 'required',
8         'role' => 'required',
9     ]), [
10        'required' => 'Kolom :attribute tidak boleh kosong',
11        'min' => 'Kolom :attribute harus memiliki minimal :min karakter',
12        'mixed' => 'Kolom :attribute harus memiliki minimal satu huruf kapital dan kecil',
13        'confirmed' => 'kolom :attribute confirmation tidak cocok',
14        'unique' => 'attribute sudah digunakan',
15        'email' => 'kolom :attribute harus ditulis dengan email yang valid'
16    ]->validate();
17    $data_pengelola = Auth()->user();
18
19    $user = user::create([
20        'name' => $request->name,
21        'uid' => Str::uuid($request->name),
22        'noHp' => $request->noHp,
23        'email' => $request->email,
24        'type' => $request->role,
25        'gender' => $request->gender,
26        'password' => Hash::make(123),
27    ]);
28    $user->remember_token = Str::random(40);
29    $user->save();
30
31    pengelola_kantor::create([
32        'id_user' => $user->id,
33        'id_kantor' => $data_pengelola->pengurusKantor()->id_kantor,
34        'admin' => 0,
35    ]);
36    Mail::to($user->email)->send(new VerificationAccountMail($user));
37
38    return redirect()->route('data.pengelola.'. Auth::user()->typeUser())->with('success', 'Akun berhasil dibuat');
39
40 }
41

```

Gambar 4. 43 Listing code create pengelola

*Listing code* pada gambar 4.43 merupakan fungsi untuk melakukan buat akun pengelola, dimana fungsi awal akan mengecek dulu menggunakan *validator* apakah data yang dimasukkan valid, lalu jika memang data sudah valid nantinya akan dibuatkan akun sesuai dengan data yang sebelumnya dimasukkan, dan langkah terakhir nantinya aplikasi akan mengirimkan email untuk melakukan verifikasi email dengan mereka akan membuat password sesuai dengan keinginan mereka.

```

1 public function deleteAkun(string $id)
2 {
3     // dd($id);
4     try {
5         $user = User::findOrFail($id);
6         $pengelola = pengelolaan_kantor::where('id_user', '=', $id);
7
8         // Hapus pengelola kantor
9         $pengelola->delete();
10
11        // Hapus user
12        $user->delete();
13
14        return redirect(route('data.pengelola.'.auth::user()->typeUser()))->with('success', 'Akun berhasil
15        dihapus');
16    } catch (ModelNotFoundException $e) {
17        return redirect('/')->with('error', 'ID tidak ditemukan di data pengelola kantor.');
```

Gambar 4. 44 Listing code delete pengelola

*Listing code* pada gambar 4.44 merupakan fungsi untuk melakukan hapus akun sesuai dengan id yang diberikan oleh *user*, dimana id tersebut didapatkan melalui kiriman dari *user*.



```

1 public function editProfilePengelola(Request $request, $id)
2 {
3     Validator::make($request->all(), [
4         'name' => 'required',
5         'email' => ['required', 'email',
6                 Rule::unique('users')->ignore($id)],
7         'gender' => 'required',
8         'noHp' => ['required', 'min:11',
9                 Rule::unique('users')->ignore($id)],
10        'role' => 'required'
11    ], [
12        'required' => 'kolom :attribute tidak boleh kosong',
13        'min' => 'kolom :attribute harus memiliki minimal :min karakter',
14        'unique' => ':attribute sudah digunakan',
15        'email' => 'kolom :attribute harus diisi dengan email yang valid!'
16    ]->validate();
17
18    $user = user::find($id);
19
20    $user->name = $request->name;
21    $user->type = $request->role;
22    $user->email = $request->email;
23    $user->save();
24
25    return redirect(route('data.pengelola.', auth::user()->typeUser()))->with('success', 'akun berhasil
diperbarui');
26 }

```

Gambar 4. 45 Listing code update data pengelola

*Listing code* pada gambar 4.45 merupakan fungsi untuk melakukan *update* data pengelola sesuai dengan data yang diberikan. Dimana akan dilakukan pengecekan terlebih dahulu terhadap inputan yang diberikan oleh *user*, jika data sudah aman maka akan dilakukan *update* sesuai data yang diberikan oleh *user* tersebut.

#### p) Fitur CRUD Kantor

```

1 public function viewKantor()
2 {
3     $kantor = kantor::get();
4     $user = auth()->user();
5
6     $data = [
7         'datatables' => $kantor,
8         'akun' => $user,
9     ];
10
11     return view('admin.kantor', $data);
12 }

```

Gambar 4. 46 Listing code read kantor

*Listing code* pada gambar 4.46 merupakan fungsi untuk melakukan *read* terhadap seluruh data kantor yang ada di *database*, yang nantinya akan digunakan untuk ditampilkan pada informasi kantor.

```

1 public function hapusKantor($id)
2 {
3     $kantor = kantor::find($id);
4
5     $kantor->proposal()->delete();
6     $kantor->pengelola_kantor()->delete();
7     $kantor->delete();
8
9     return redirect(route('admin.viewKantor'))->with('success', 'Berhasil menghapus kantor!');
10 }

```

Gambar 4. 47 Listing code delete kantor

*Listing code* pada gambar 4.47 merupakan fungsi untuk melakukan *delete* terhadap data kantor, yang dimana nantinya

akan otomatis juga menghapus seluruh data proposal dan juga pengelola kantor.

```
1 public function updateKantor(Request $request, $id)
2 {
3     $kantor = Kantor::find($id);
4     $kantor->update([
5         'name' => $request->NamaKantor,
6         'email' => $request->EmailKantor,
7         'noWa' => $request->NoKantor,
8         'alamat' => $request->AlamatKantor
9     ]);
10
11     return redirect(route('admin.viewKantor'));
12 }
```

Gambar 4. 48 Listing code update kantor

*Listing code* pada gambar 4.48 merupakan fungsi untuk melakukan *update* terhadap data kantor, yang diawali dengan pencarian id yang sesuai dengan yang diberikan oleh *user* dan setelah itu akan dilakukan proses *update* sesuai dengan yang diinputkan oleh *user* tersebut.

```
1 public function buatKantor(Request $request)
2 {
3     Validator::make($request->all(), [
4         'name' => 'required',
5         'email' => 'required|email|unique:kantors,email',
6         'alamat' => 'required',
7         'noWa' => 'required|min:11|unique:kantors,noWa'
8     ], [
9         'required' => 'Kolom :attribute tidak boleh kosong',
10        'min' => 'Kolom :attribute harus memiliki minimal :min karakter',
11        'mixed' => 'Kolom :attribute harus memiliki minimal satu huruf kapital dan kecil',
12        'confirmed' => 'kolom :attribute confirmation tidak cocok',
13        'unique' => 'attribute sudah digunakan',
14        'email' => 'kolom :attribute harus diisi dengan email yang valid!'
15    ]->validate();
16
17    kantor::create([
18        'name' => $request->name,
19        'email' => $request->email,
20        'alamat' => $request->alamat,
21        'noWa' => $request->noWa
22    ]);
23
24    return redirect(route('admin.viewKantor'))->with('success', 'kantor berhasil dibuat!');
25
26 }
```

Gambar 4. 49 Listing code create kantor

*Listing code* pada gambar 4.49 merupakan fungsi untuk melakukan buat kantor, yang dimana akan diawali dengan pengecekan menggunakan *validator* supaya data yang dimasukan oleh *user* sesuai, dan jika data sudah sesuai semua maka kantor akan dibuat dan dapat digunakan untuk menampung para pengelola lazismu.

#### 4) *Testing* atau Pengujian Aplikasi

Setelah sistem aplikasi berhasil diselesaikan, selanjutnya dilakukan pengukuran tingkat usability sistem menggunakan metode *System Usability Scale* (SUS). Pengukuran ini dilakukan untuk mengetahui apakah sistem tersebut dapat digunakan dengan mudah dan efektif oleh pengguna.

#### c) **Melakukan pengujian aplikasi menggunakan *system usability scale***

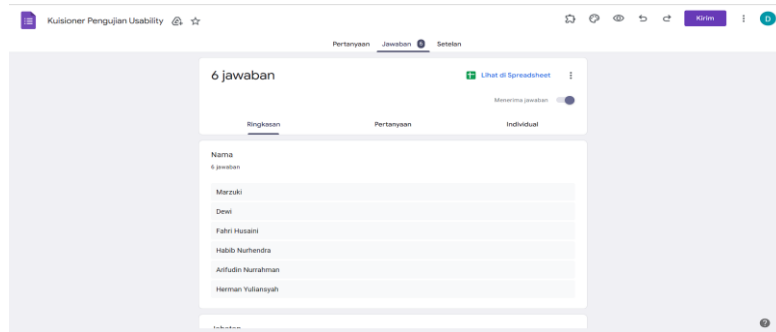
Pengujian dilakukan dengan menggunakan metode SUS, kepada 6 pengguna dengan level *user* yang berbeda-beda. Dimana terdapat 10 pertanyaan dengan skala jawaban 1-5, dan untuk detail dari isian pertanyaan dapat dilihat pada tabel 4.1.

##### 1) Daftar Pertanyaan

Tabel 4. 1 Daftar pertanyaan

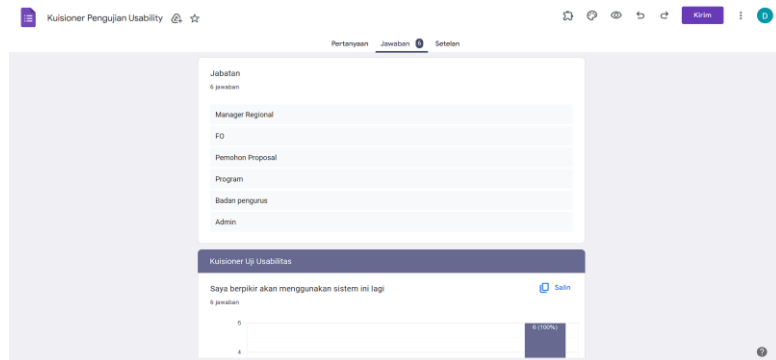
| No | Pertanyaan   |
|----|--|
| 1  | Saya berpikir akan menggunakan sistem ini lagi                                     |
| 2  | Saya merasa sistem ini rumit untuk digunakan                                       |
| 3  | Saya merasa sistem ini mudah digunakan   |
| 4  | Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini |
| 5  | Saya merasa fitur-fitur sistem ini berjalan dengan semestinya                      |
| 6  | Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)     |
| 7  | Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat      |
| 8  | Saya merasa sistem ini membingungkan   |
| 9  | Saya merasa tidak ada hambatan dalam menggunakan sistem ini                        |
| 10 | Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini         |

## 2) Hasil Pengujian



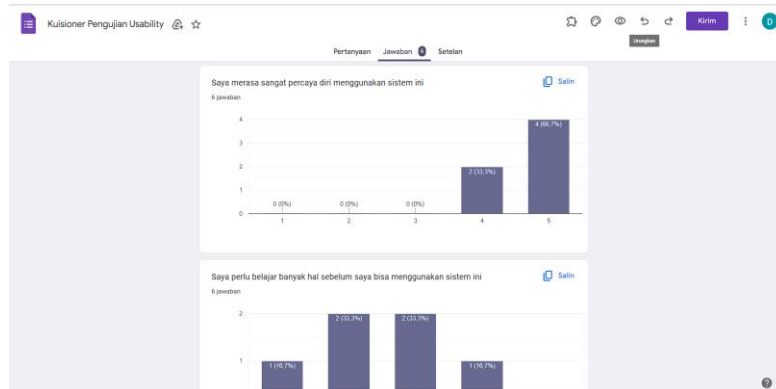
Gambar 4. 50 Nama responden

Pada gambar 4.41 merupakan nama dari para responden saat pengujian *system usability scale* berlangsung.



Gambar 4. 51 Jabatan / role responden

Pada gambar 4.42 merupakan jabatan dari masing masing responden.



Gambar 4. 52 Nilai kepuasan *user* terhadap aplikasi

Pada gambar 4.43 merupakan hasil perolehan dari tingkat nilai kepuasan yang dirasakan oleh para *user* terhadap aplikasi ini.

3) Analisis hasil pengujian

Metode perhitungan:

1. Tiap pertanyaan bernomor ganjil, maka skor dari tiap pertanyaan tersebut dikurangi 1.
2. Tiap pertanyaan bernomor genap, maka 5 dikurangi dengan skor dari tiap pertanyaan tersebut.

Tabel 4. 2 Data hasil pengujian aplikasi

| Partisipan | q1 | q2 | q3 | q4 | q5 | q6 | q7 | q8 | q9 | q10 | Skor SUS |
|------------|----|----|----|----|----|----|----|----|----|-----|----------|
| p1         | 5  | 2  | 5  | 2  | 5  | 2  | 4  | 2  | 4  | 4   | 77.5     |
| p2         | 5  | 2  | 4  | 5  | 4  | 2  | 5  | 2  | 4  | 3   | 67.5     |
| p3         | 5  | 2  | 4  | 1  | 5  | 1  | 5  | 2  | 5  | 2   | 92.5     |
| p4         | 5  | 2  | 4  | 2  | 4  | 2  | 5  | 2  | 5  | 1   | 85       |
| p5         | 5  | 3  | 4  | 3  | 5  | 1  | 4  | 3  | 5  | 3   | 75       |
| p6         | 5  | 1  | 4  | 2  | 5  | 2  | 5  | 2  | 5  | 2   | 87.5     |

Pada tabel 4.2 merupakan data hasil pengujian aplikasi yang telah dilakukan sebelumnya oleh para responden dari lazizmu, dan setelah mendapatkan data diatas maka akan dihitung untuk nilai rata-rata dari ke 4 responden tersebut dengan rumus seperti berikut.

$$\begin{aligned}
 \text{Rata-rata} &= (77.5+67.5+92.5+85+75+87.5) / 6 \\
 &= 485 / 6 \\
 &= 80.83
 \end{aligned}$$

4) Kesimpulan hasil pengujian

Tabel 4. 3 Data hasil *system usability scale*

| No | Kategori              | Rentang Skor | Jumlah | Persentase |
|----|-----------------------|--------------|--------|------------|
| 1  | <i>Not Acceptable</i> | 0-50         | 0      | 0%         |
| 2  | Marginal              | 50-70        | 1      | 16.67%     |
| 3  | <i>Acceptable</i>     | 70-100       | 5      | 83.33%     |

Berdasarkan pada *table* 4.3 didapatkan nilai rata-rata dari responden merupakan 80.83 dengan rincian sebagai berikut:

- 1) 16.67% responden memberi penilaian marginal
- 2) 83.33% responden menyatakan sistem/aplikasi bisa diterima (*acceptable*)

Berdasar rerata penilaian SUS sebesar 80.83, maka sistem/aplikasi dinyatakan dapat diterima (*acceptable*).

**d) Realisasi Jadwal Kegiatan Magang**

Praktik magang dilaksanakan selama 3 bulan, dari tanggal 1 Oktober 2023 hingga 30 Desember 2023. Berikut adalah rincian jadwal realisasi praktik magang yang telah dirancang sebelumnya di Lazismu DIY :

Tabel 4. 4 Jadwal realisasi kegiatan magang

| No | Nama Kegiatan  | Oktober |   |   |   | November |   |   |   | Desember |   |   |   | Realisasi    |      |
|----|--|---------|---|---|---|----------|---|---|---|----------|---|---|---|--------------|------|
|    |  | 1       | 2 | 3 | 4 | 1        | 2 | 3 | 4 | 1        | 2 | 3 | 4 | Ya/<br>Tidak | %    |
| 1  | Pemahaman terhadap kebutuhan system  | ■       |   |   |   |          |   |   |   |          |   |   |   | Ya           | 100% |
| 2  | Pemahaman terhadap <i>framework laravel</i>  | ■       | ■ |   |   |          |   |   |   |          |   |   |   | Ya           | 100% |
| 3  | Pembuatan rancangan <i>relational database management system</i>   | ■       | ■ |   |   |          |   |   |   |          |   |   |   | Ya           | 100% |
| 4  | Implementasi <i>relational database management system</i> , kedalam kode <i>laravel (migrations &amp; seeders)</i> |         |   | ■ | ■ |          |   |   |   |          |   |   |   | Ya           | 100% |
| 5  | Pembuatan fungsionalitas untuk <i>user pemohon</i>   |         |   |   |   | ■        | ■ |   |   |          |   |   |   | Ya           | 100% |
| 6  | Pembuatan fungsionalitas untuk <i>user front office</i> serta perbaikan bug minor                                  |         |   |   |   |          | ■ | ■ |   |          |   |   |   | Ya           | 100% |
| 7  | Pembuatan fungsionalitas untuk   |         |   |   |   |          |   | ■ | ■ |          |   |   |   | Ya           | 100% |

|    |   |  |  |  |  |  |  |  |  |  |  |  |    |      |
|----|---|--|--|--|--|--|--|--|--|--|--|--|----|------|
|    | <i>user</i> manager serta perbaikan bug minor                                       |  |  |  |  |  |  |  |  |  |  |  |    |      |
| 8  | Pembuatan fungsionalitas untuk <i>user</i> badan pengurus serta perbaikan bug minor |  |  |  |  |  |  |  |  |  |  |  | Ya | 100% |
| 9  | Pembuatan fungsionalitas untuk <i>user</i> program serta perbaikan bug minor        |  |  |  |  |  |  |  |  |  |  |  | Ya | 100% |
| 10 | Pembuatan fungsionalitas untuk <i>user</i> admin serta perbaikan bug minor          |  |  |  |  |  |  |  |  |  |  |  | Ya | 100% |
| 11 | Melakukan testing aplikasi ke client  |  |  |  |  |  |  |  |  |  |  |  | Ya | 100% |



#### **4. Keberlanjutan**

Praktik magang di Lazismu DIY bersifat tidak berkelanjutan dan hanya dilaksanakan selama periode program.

## **BAB IV**

### **PENUTUP**

#### **A. Kesimpulan**

Perancangan Sistem Aplikasi Pengajuan Proposal dengan menggunakan framework Laravel berbasis website di Lazismu DIY telah selesai dilakukan. Berikut adalah kesimpulannya.

1. Sistem back-end pada aplikasi pengajuan proposal di lazismu berhasil dikembangkan dengan menggunakan framework laravel dan metode pengembangan agile.

#### **B. Saran**

Sistem aplikasi pengajuan proposal yang telah dibangun masih memiliki beberapa kekurangan, baik dari segi tampilan maupun fiturnya. Untuk itu, sistem ini perlu dikembangkan lebih lanjut untuk memenuhi kebutuhan pengguna. Berikut adalah beberapa saran yang dapat diberikan sebagai acuan pengembangan sistem aplikasi pengajuan proposal:

1. Menambahkan table history proposal pada database. Table ini berguna untuk menyimpan proposal yang telah selesai (disetujui atau ditolak). Dengan adanya table ini, maka Lazismu DIY dapat dengan mudah melacak status proposal yang telah diajukan, serta dapat digunakan untuk keperluan analisis dan evaluasi.

## LAMPIRAN

### A. Surat Permohonan Izin Praktik Magang

|   | <b>UNIVERSITAS AHMAD DAHLAN</b><br><b>FAKULTAS TEKNOLOGI INDUSTRI</b><br>KAMPUS 4 : Jl. Ahmad Yani, Tamanan, Banguntapan, Bantul, Yogyakarta. 55191<br>☎ 0274 563515, 511830 ext 41211 📠 0819 9079 0454 🌐 ftl.uad.ac.id ✉ ftl@uad.ac.id |            |     |    |                     |            |  |
|--|---|------------|-----|----|---------------------|------------|--|
| Nomor : F2.1/124/D.33/IX/2023<br>Hal : Permohonan Izin Kerja   | 22 September 2023   |            |     |    |                     |            |  |
| Kepada<br>Yth. Direktur<br><b>LazizMu DIY</b><br>Jl. Gedongkuning No.152, RT.41,<br>Rejowinangun, Kotagede, Yogyakarta, Daerah Istimewa Yogyakarta 55171   |   |            |     |    |                     |            |  |
| Dengan hormat,   |   |            |     |    |                     |            |  |
| Dalam rangka memberikan kesempatan kepada mahasiswa dalam mengimplementasikan disiplin ilmu yang diperoleh di bangku kuliah dengan keadaan sebenarnya, Pimpinan Fakultas Teknologi Industri mengajukan permohonan kepada Bapak/Ibu untuk berkenan memberikan ijin kepada mahasiswa kami tersebut di bawah ini untuk melaksanakan <i>Kerja</i> di instansi yang Bapak/Ibu pimpin. |   |            |     |    |                     |            |  |
| Adapun data mahasiswa sebagai berikut:   |   |            |     |    |                     |            |  |
| <table border="1"><thead><tr><th>No</th><th>Nama</th><th>NIM</th></tr></thead><tbody><tr><td>1.</td><td>Dewa Putra Hernanda</td><td>2000018270</td></tr></tbody></table>   | No  | Nama       | NIM | 1. | Dewa Putra Hernanda | 2000018270 |  |
| No   | Nama  | NIM        |     |    |                     |            |  |
| 1.   | Dewa Putra Hernanda   | 2000018270 |     |    |                     |            |  |
| Program Studi : <b>S1 Informatika</b><br>Fakultas : Teknologi Industri   |   |            |     |    |                     |            |  |
| Sebagai bahan pertimbangan Bapak/Ibu kami sampaikan bahwa sebagai konsekuensinya, mahasiswa yang bersangkutan bersedia memenuhi persyaratan administratif yang diperlukan.   |   |            |     |    |                     |            |  |
| Waktu Pelaksanaan Kerja Praktek Mulai Tanggal 1 Oktober - 31 Desember 2023.  |   |            |     |    |                     |            |  |
| Atas perhatian dan kerjasama Bapak/Ibu, kami sampaikan terima kasih.   |   |            |     |    |                     |            |  |
|  | Hormat kami,<br>a.n Dekan<br>Wakil Dekan<br><br>r. Sri Winiarti, S.T., M.Cs.<br>NIPM.19751216 200103 011 0880702                                    |            |     |    |                     |            |  |

B. Sertifikat Praktik Magang



C. Form Penilaian Pembimbing Lapangan


**FORM PENILAIAN PEMBIMBING LAPANGAN**

Yang bertanda tangan di bawah ini :  
 Nama : MARZULI  
 NIK / NIP : .....  
 Jabatan : MANAGER PESIWA

Sebagai Pembimbing Lapangan Praktik Magang :  
 Nama : Dewa Putra Hernanda  
 NIM : 2000018270

Setelah mengikuti pelaksanaan Praktik Magang mahasiswa tersebut, memberikan nilai sebagai berikut :

| ASPEK PENILAIAN                                   | Parameter Penilaian |           |            |             |            | Skor      |
|---|---------------------|-----------|------------|-------------|------------|-----------|
|   | Sangat Baik<br>5    | Baik<br>4 | Cukup<br>3 | Kurang<br>2 | Buruk<br>1 |           |
| 1. Komunikasi dan Pemahaman Instruksi Kerja       |                     | ✓         |            |             |            | 4         |
| 2. Keaktifan dan Respon di Tempat Kerja           |                     | ✓         |            |             |            | 4         |
| 3. Kesesuaian Materi KP dengan Keilmuan Mahasiswa |                     | ✓         |            |             |            | 4         |
| 4. Disiplin Kerja                                 |                     | ✓         |            |             |            | 4         |
| 5. Pengerjaan Tugas dan Ketrampilan Teknis        |                     | ✓         |            |             |            | 4         |
| <b>Total</b>                                      |                     |           |            |             |            | <b>20</b> |

Yogyakarta, 5 Januari 2024  
 Pembimbing Lapangan\*  
  
 MARZULI  
 NIP: .....

\*Pengisian lapangan harus diandatangani dan dibubuhi cap

## D. Logbook

### Sebelum Uts



**LOG BOOK PRAKTIK MAGANG MAHASISWA  
PROGRAM STUDI S1 INFORMATIKA T.A 2020 / 2021  
(WAJIB DIISI DAN MASUK DALAM PENILAIAN)**



Nim : 2000018270  
 Nama Mahasiswa : Dewa Putra Hernanda  
 Judul Praktikum Magang : Website Pengajuan Proposal  
 Dosen Pembimbing : Herman Yuliansyah, s.t., m.eng., ph.d.  
 Pembimbing Lapangan : marzuki, SE



**Petunjuk Pengisian Log Book**



1. Log book di isi per minggu
2. Log book ditulis tangan
3. Setiap kegiatan di paraf oleh pembimbing lapangan/ dosen pembimbing Praktikum Magang
4. Log book per minggu di paraf oleh dosen pengampu kelas Praktikum Magang
5. Jumlah blimbingan minimal 4x

**Logbook Minggu 4 sd 7 (sebelum UTS)**

| No | Kegiatan dan Lokasi Praktikum Magang   | Waktu Pelaksanaan                         |                           | Hasil   | Kendala, Rencana Perubahan (jika ada) | Paraf Pembimbing Lapangan   | Paraf Dosen Pembimbing Praktikum Magang   |
|----|--|---|---------------------------|---|---------------------------------------|---|---|
|    |  | Hari/TGL                                  | Jam Durasi                |   |                                       |   |   |
| 1  | <b>Lazimu DIY</b><br>- Diskusi mengenai permasalahan dalam sistem pengajuan proposal di lazimu DIY.<br>- Pembuatan Github untuk melakukan collaboration dalam pengembangan website.<br>- Menentukan timeline project & pembagian jobdesk | Minggu 1,<br>25 – 30<br>September<br>2023 | 4 x 6<br>hari =<br>24 Jam | - Mengetahui permasalahan utama dalam pengajuan proposal di lazimu, dan kami diharapkan untuk membuat website yang dapat melakukan pengajuan proposal secara digital.<br>- Telah dibuatkan dan diinvite pada satu repository github yang nantinya akan dikembangkan menjadi website pengajuan proposal. |                                       |  |  |

|   |  |                                       |                           |  |  |  |   |
|---|--|---------------------------------------|---------------------------|--|--|--|---|
| 2 | Menganalisis kebutuhan website secara lebih detail dibagian:<br>- Proses bisnis<br>- Requirement website<br>- ERD<br>- Rancangan Database<br>- Memahami alur kerja laravel | Minggu 2,<br>2 – 7<br>Oktober<br>2023 | 4 x 6<br>hari =<br>24 Jam | - Mengetahui timeline serta jobdesk yang diberikan kepada saya<br>- Proses bisnis telah berhasil dirancang serta dipahami guna mencapai tujuan yang ditetapkan, meliputi interaksi dengan pengguna antara pemohon dan juga pengelola, serta flow dalam pemrosesan data.<br>- Kebutuhan serta spesifikasi website telah ditentukan, termasuk dengan fitur, fungsionalitas, tampilan, dan pengalaman pengguna yang diinginkan.<br>- ERD telah dibuat guna untuk menggambarkan struktur basis data dan hubungan entitas dengan atributnya.<br>- Rancangan database telah berhasil dibuat sesuai dengan flow yang dibutuhkan dalam pengembangan aplikasi web pengajuan proposal.<br>- Telah memahami alur kerja framework laravel supaya dapat lebih efektif dan efisien dalam pengembangan aplikasi web pengajuan proposal. |  |  |  |
|---|--|---------------------------------------|---------------------------|--|--|--|---|

|   |   |  |                                     |   |  |   |  |
|---|---|--|-------------------------------------|---|--|---|--|
| 3 | <p>Mulai membuat aplikasi dengan jobdesk seperti:</p> <ul style="list-style-type: none"> <li>- Database</li> <li>- Seeders</li> <li>- Authentication</li> <li>- Upload proposal</li> <li>- Edit Proposal</li> </ul> | <p>Minggu 3 &amp; 4,<br/>9 - 21<br/>Oktober<br/>2023</p> | <p>4 x 12<br/>hari =<br/>48 jam</p> | <p><b>Database</b></p> <ul style="list-style-type: none"> <li>- Membuat migrations database di-apikasi sesuai dengan rancangan database yang sudah ditetapkan minggu lalu</li> </ul> <p><b>Seeders</b></p> <ul style="list-style-type: none"> <li>- Membuat data template awal saat database dibuat, supaya dapat memudahkan dalam tahap maintenance kedepannya</li> </ul> <p><b>Authentication</b></p> <ul style="list-style-type: none"> <li>- Membuat 5 role utama yaitu pemohon, front office, manager, badan pengusaha, admin, yang nantinya jika login role tersebut memiliki hak akses yang berbeda - beda</li> </ul> <p><b>Upload proposal</b></p> <ul style="list-style-type: none"> <li>- Membuat method untuk user pemohon dapat mengirimkan proposal yang nantinya diberikan kepada pengelola (front-office/badan pengusaha/manager)</li> </ul> <p><b>Edit proposal</b></p> <ul style="list-style-type: none"> <li>- Membuat method untuk user pemohon jika status proposal masih dalam tahap diusulkan maka pemohon dapat melakukan edit data proposal.</li> </ul> |  | <p></p> | <p></p> |
|---|---|--|-------------------------------------|---|--|---|--|

|   |   |  |                                     |  |  |  |   |
|---|---|--|-------------------------------------|--|--|--|---|
| 4 | <p>Membuat fitur untuk pemohon :</p> <ul style="list-style-type: none"> <li>- Membuat akun</li> <li>- Menampilkan data proposal yang telah diajukan</li> <li>- Update data diri</li> <li>- Melihat riwayat proposal</li> <li>- Mencari proposal berdasarkan kode tracking yang diberikan</li> </ul> | <p>Minggu 5 &amp; 6, 23<br/>Oktober -4<br/>November<br/>2023</p> | <p>4 x 12<br/>hari =<br/>48 jam</p> | <p><b>Membuat akun</b></p> <ul style="list-style-type: none"> <li>- Membuat method untuk create akun bagi pemohon yang ingin melakukan upload proposal secara mandiri</li> </ul> <p><b>Menampilkan data proposal yang telah diajukan</b></p> <ul style="list-style-type: none"> <li>- Menampilkan data proposal yang telah diajukan oleh pemohon jika status masih diusulkan, diverifikasi, atau menunggu persetujuan)</li> </ul> <p><b>Update data diri</b></p> <ul style="list-style-type: none"> <li>- User pemohon dapat melengkapi data diri sebelum bisa melakukan upload proposal</li> </ul> <p><b>Melihat riwayat proposal</b></p> <ul style="list-style-type: none"> <li>- User pemohon dapat melihat riwayat proposal, dengan status proposal yang telah berubah menjadi diterima/ditolak</li> </ul> <p><b>Mencari proposal berdasarkan kode tracking yang diberikan</b></p> <ul style="list-style-type: none"> <li>- User pemohon dapat mencari dan melihat status proposal yang mereka ajukan dengan cara mengirimkan kode tracking yang diberikan kepada pemohon</li> </ul> |  | <p></p> | <p></p> |
|---|---|--|-------------------------------------|--|--|--|---|

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|

Catatan Pembimbing Lapangan/Dosen Pembimbing Praktik Magang / Dosen Pengampu Kelas Praktik Magang:

.....  
.....  
.....  
.....

Dosen Pengampu Kelas Praktik Magang

Yogyakarta, 7 November 2023  
Mahasiswa

  
(Guntur Maulana Zamroni, B.Sc., M.Kom.)  
NIDN.0509038402

  
(Dewa Putra Hernanda)  
NIM. 2000018270

## Setelah Uts

### LOG BOOK PRAKTIK MAGANG MAHASISWA PROGRAM STUDI S1 INFORMATIKA, UAD T.A 2020 / 2021 (WAJIB DIISI DAN MASUK DALAM PENILAIAN)

Nim : 2000018270  
 Nama Mahasiswa : Dewa Putra Hemanda  
 Judul Praktik Magang : Website Pengajuan Proposal  
 Dosen Pembimbing : Herman Yuliansyah, s.t., m.eng., ph.d.  
 Pembimbing Lapangan : marzuki, SE

#### Petunjuk Pengisian Log Book





1. Log book di isi per minggu
2. Log book ditulis tangan
3. Setiap kegiatan di paraf oleh pembimbing lapangan/ dosen pembimbing Praktik Magang
4. Log book per minggu di paraf oleh dosen pengampu kelas Praktik Magang
5. Jumlah bimbingan minimal 3x

#### Logbook Minggu 8 sd 10 (setelah UTS)

| No | Kegiatan dan Lokasi Praktik Magang   | Waktu Pelaksanaan                           |                     | Hasil  | Kendala, Rencana Perubahan (Jika ada) | Paraf Pembimbing Lapangan | Paraf Dosen Pembimbing Praktik Magang |
|----|--|---|---------------------|--|---------------------------------------|---------------------------|---------------------------------------|
|    |  | Hari/TGL                                    | Jam Durasi          |  |                                       |                           |                                       |
| 5  | Membuat fitur user manager<br>- Menyelaraskan persepsi tentang alur tugas manager<br>- Membuat fungsi untuk user Manager | Minggu 7 dan Minggu 8, 6 - 18 November 2023 | 12 x 4 jam = 48 jam | Menyelaraskan persepsi tentang alur tugas manager<br>Hasil : Seluruh team sudah paham dengan tugas manager yang sudah dijelaskan oleh client<br><br>Membuat fungsi untuk user Manager<br>Hasil : Function pada controller untuk melakukan CRUD pada bagian<br>- Create, Update, Delete, Read akun untuk pengelola baru<br>- Read dashboard kantor<br>- Read history proposal<br>- Read proposal<br>- Update profile user |                                       |                           |                                       |

|   |   |  |                    |  |  |  |  |
|---|---|--|--------------------|--|--|--|--|
| 6 | Membuat fitur user Badan Pengurus<br>- Menyelaraskan persepsi tentang alur tugas Badan Pengurus<br>- Membuat fungsi untuk user Badan Pengurus | Minggu 9, 20 - 25 November 2023          | 6 x 4 jam = 24 jam | - Dan melakukan verifikasi proposal untuk bagian manager<br><br>Menyelaraskan persepsi tentang alur tugas Badan Pengurus.<br>Hasil : Seluruh team sudah paham dengan tugas Badan Pengurus yang sudah dijelaskan oleh client<br><br>Membuat fungsi untuk user Badan Pengurus<br>Hasil : Function pada controller untuk melakukan pada bagian<br>- Read dashboard proposal<br>- Verifikasi proposal untuk bagian badan pengurus<br>- Update profile user |  |  |  |
| 7 | Membuat fitur user program<br>- Menyelaraskan persepsi tentang alur tugas program.<br>- Membuat fungsi untuk user Program                     | Minggu 10, 27 November - 2 Desember 2023 | 6 x 4 jam = 24 jam | Menyelaraskan persepsi tentang alur tugas program.<br>Hasil : Seluruh team sudah paham dengan tugas program yang sudah dijelaskan oleh client<br><br>Membuat fungsi untuk user Program<br>Hasil : Function pada controller untuk melakukan pada bagian<br>- Read dashboard proposal<br>- Verifikasi proposal untuk bagian program<br>- Update profile user   |  |  |  |



|   |   |                                  |                    |   |  |  |   |
|---|---|----------------------------------|--------------------|---|--|--|---|
| 8 | Membuat fitur user super admin<br>- Menyelaraskan persepsi tentang alur tugas super admin.<br>- Membuat fungsi untuk user super admin | Minggu 11, 4 – 9 Desember 2023   | 6 x 4 jam = 24 jam | Menyelaraskan persepsi tentang alur tugas super admin.<br>Hasil : Seluruh team sudah paham dengan tugas super admin yang sudah dijelaskan oleh client<br><br>Membuat fungsi untuk user super admin<br>Hasil : Function pada controller untuk melakukan pada bagian<br>- CRUD Kantor<br>- CRUD Pengelola kantor        |  |  |  |
| 9 | Membuat testing SuS dan diujikan<br>- Membuat testing SuS dengan gform<br>- Menguji ke user kantor Lazismu DIY                        | Minggu 12, 11 – 16 Desember 2023 | 6 x 6 jam = 36 jam | Membuat Testing SuS dengan Google Forms<br>Hasil: Terbentuknya kuesioner SuS yang dapat digunakan untuk mengumpulkan data tentang usability sistem.<br><br>Menguji ke User Kantor Lazismu DIY<br>Hasil: Kumpulan data dari hasil pengujian SuS yang mencakup respons dan penilaian pengguna terhadap kegunaan sistem. |  |  |  |

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|

Catatan Pembimbing Lapangan/Dosen Pembimbing Praktik Magang / Dosen Pengampu Kelas Praktik Magang:

.....

.....

.....

Dosen Pengampu Kelas Praktik Magang

Yogyakarta, 3 Januari 2023  
Mahasiswa



(Guntur Maulana Zamrodji, B.Sc., M.Kom.)  
NIDN.0509038402



(Dewa Putra Hernanda)  
NIM. 2000018270

E. Dokumentasi Kegiatan Praktik Magang

1. Suasana dan ruangan magang di Lazismu DIY



2. Kegiatan Praktik Magang



### 3. Presentasi progress project aplikasi

